4 Select a bit depth to set the number of bits per pixel and colors to use in creating the image:

- Select 8-bit for a 256-color image.
- Select 24-bit for thousands of colors.
- Select 24-bit with Alpha for thousands of colors with transparency (32 bits).

The higher the bit depth, the larger the file.

5 Select options to specify appearance settings for the exported PNG:

Optimize Colors removes any unused colors from a PNG file's color table. This option reduces the file size by 1000 to 1500 bytes without affecting image quality but increases the memory requirements slightly. This option has no effect on an adaptive palette.

Interlace incrementally displays the exported PNG in a browser as it downloads. Interlacing lets the user see basic graphic content before the file has completely downloaded and may download the file faster over a slow network connection. Do not interlace an animated PNG file.

Smooth applies anti-aliasing to an exported bitmap to produce a higher-quality bitmap image and improve text display quality. However, smoothing may cause a halo of gray pixels to appear around an anti-aliased image placed on a colored background, and it increases the PNG file size. Export an image without smoothing if a halo appears or if you're placing a PNG transparency on a multicolored background.

Dither Solids applies dithering to solid colors and gradients. See Dither options in step 6.

Remove Gradients, turned off by default, converts all gradient fills in the application to solid colors using the first color in the gradient. Gradients increase the size of a PNG and often are of poor quality. If you use this option, select the first color of your gradients carefully to prevent unexpected results.

6 If you chose 8-bit for Bit Depth in step 4, select a Dither option to specify how pixels of available colors are mixed to simulate colors not available in the current palette. Dithering can improve color quality, but it increases file size. Select from the following options:

None turns off dithering and replaces colors not in the basic color table with the solid color from the table that most closely approximates the specified color. Turning dithering off can produce smaller files but unsatisfactory colors.

Ordered provides good-quality dithering with the smallest increase in file size.

Diffusion provides the best-quality dithering but increases file size and processing time. It also works only with the web 216 color palette selected.

7 Select Palette Type to define the color palette for the PNG image:

Web 216 uses the standard 216-color, browser-safe palette to create the PNG image, for good image quality and the fastest processing on the server.

Adaptive analyzes the colors in the image and creates a unique color table for the selected PNG file. This option is best for systems displaying thousands or millions of colors; it creates the most accurate color for the image but results in a file size larger than a PNG created with the web 216 palette.

Web Snap Adaptive is the same as the Adaptive palette option except that it converts very similar colors to the web 216 color palette. The resulting color palette is optimized for the image, but when possible, Flash uses colors from web 216. This produces better colors for the image when the web 216 palette is active on a 256-color system.

To reduce the size of a PNG file with an adaptive palette, use the Max Colors option to decrease the number of palette colors, as described in the next step.

Custom specifies a palette that you have optimized for the selected image. The custom palette is processed at the same speed as the web 216 palette. To use this option, you should know how to create and use custom palettes. To select a custom palette, click the Ellipsis (...) button to the right of the Palette box at the bottom of the dialog box and select a palette file. Flash supports palettes saved in the ACT format, exported by Macromedia Fireworks and other leading graphics applications; for more information, see "Importing and exporting color palettes" on page 75.

8 If you selected the Adaptive or Web Snap Adaptive palette in step 7, enter a value for Max Colors to set the number of colors used in the PNG image. Choosing a smaller number of colors can produce a smaller file but may degrade the colors in the image.

9 Select Filter Options to select a line-by-line filtering method to make the PNG file more compressible, and experiment with the different options for a particular image:

None turns off filtering.

Sub transmits the difference between each byte and the value of the corresponding byte of the prior pixel.

Up transmits the difference between each byte and the value of the corresponding byte of the pixel immediately above.

Average uses the average of the two neighboring pixels (left and above) to predict the value of a pixel.

Path computes a simple linear function of the three neighboring pixels (left, above, upper left), and then chooses as a predictor the neighboring pixel closest to the computed value.

Adaptive analyzes the colors in the image and creates a unique color table for the selected PNG file. This option is best for systems displaying thousands or millions of colors; it creates the most accurate color for the image but results in a file size larger than a PNG created with the web 216 palette. You can reduce the size of a PNG created with an adaptive palette by decreasing the number of colors in the palette.

10 To save the settings with the current file, click OK.

Specifying publish settings for QuickTime movies

The QuickTime Publish Settings option creates movies in the same QuickTime format you have installed on your computer. For example, if you have QuickTime 5 installed, Flash publishes the QuickTime movie in version 5.

The Flash document plays in the QuickTime movie exactly as it does in Flash Player, retaining all of its interactive features. If the Flash document also contains a QuickTime movie, Flash copies it to its own track in the new QuickTime file.

The current version of the QuickTime Player (as of this writing) supports playback of Flash Player 4 SWF files. For best results, Flash content that you export to the QuickTime format should contain only those features supported by Flash Player 4. Future releases of the QuickTime Player may support additional Flash file formats.

If you try to export Flash Player 6 or 7 content to the QuickTime format, an error message will appear indicating that the installed version of QuickTime does not support that version of Flash Player.

To resolve this issue, you can select Flash Player 4 from the Version pop-up menu on the Flash tab of the Publish Settings dialog box. For more information, see "Setting publish options for the Flash SWF file format" on page 282.

If a newer version of the QuickTime Player becomes available that supports Flash Player 6 and later versions, you'll be able to install the updated QuickTime version and publish your document as QuickTime files that target those versions of Flash Player.

For more information on QuickTime movies, see your QuickTime documentation.

To publish a QuickTime movie with your Flash SWF file:

1 Do one of the following to open the Publish Settings dialog box:

- Select File > Publish Settings.
- In the Property inspector for the document (which is available when no object is selected), click the Settings button.

Note: To create a publish profile for the publish settings that you'll specify, see "Using publish profiles" on page 295.

2 On the Formats tab, select the QuickTime file type. For the QuickTime filename, either use the default filename, or enter a new filename with the .mov extension.

3 Click the QuickTime panel to display its settings.

4 For Dimensions, enter a width and height in pixels for the exported QuickTime movie, or select Match Movie to make the QuickTime movie the same size as the Flash SWF file and keep its aspect ratio.

5 Select an Alpha option to control the transparency (alpha) mode of the Flash track in the QuickTime movie without affecting any alpha settings in the Flash application:

Alpha Transparent makes the Flash track SWF file transparent and shows any content in tracks behind the Flash track.

Copy makes the Flash track opaque and masks all content in tracks behind the Flash track.

Auto makes the Flash track transparent if it is on top of any other tracks, but opaque if it is the bottom or only track in the SWF file.

6 Select a Layer option to control where the Flash track plays in the stacking order of the QuickTime movie:

Top places the Flash track always on top of other tracks in the QuickTime movie.

Bottom places the Flash track always behind other tracks.

Auto places the Flash track in front of other tracks if Flash objects are in front of video objects in the Flash application, and behind all other tracks if Flash objects are not in front.

7 Select Streaming Sound to have Flash export all the streaming audio in the Flash SWF file to a QuickTime sound track, recompressing the audio using the standard QuickTime audio settings. To change these options, click Audio Settings; for more information, see your QuickTime documentation.

8 Select Controller to specify the type of QuickTime controller used to play the exported movie—None, Standard, or QuickTime VR.

9 Select Playback options to control how QuickTime plays a movie:

Looping repeats the movie when it reaches the last frame.

Paused at Start pauses the movie until a user clicks a button in the movie or selects Play from the shortcut menu. By default, the option is deselected; that is, the movie begins to play as soon as it is loaded.

Play Every Frame displays every frame of the movie without skipping to maintain time and does not play sound.

10 Select File Flatten (Make Self-Contained) to combine the Flash content and imported video content into a single QuickTime movie. Deselecting this option makes the QuickTime movie refer to the imported files externally; the movie won't work properly if these files are missing.

11 To save the settings with the current file, click OK.

Using publish profiles

You can create a publish profile that saves a configuration of publish settings. You can then export the publish profile for use in other documents, or for use by others. Conversely, you can import publish profiles for use in your document. Publish profiles offer many advantages, including the following:

- You can create profiles to publish in a variety of media formats.
- You can create a publish profile for an in-house way of publishing your files that differs from the way you'd publish the files for a client.
- Your company can create a standard publish profile to ensure files are published uniformly.

Publish profiles, like default publish settings, are saved at the document rather than application level. To use a publish profile in another document, you export it, then import it into the other file. See "Exporting a publish profile" on page 296 and "Importing a publish profile" on page 296.

Creating a publish profile

The Publish Settings dialog box includes a Create New Profile button, which creates a profile based on the publish settings you've specified.

To create a publish profile:

1 In the Publish Settings dialog box, click the Create New Profile (+) button.

2 In the Create New Profile dialog box, name the publish profile and click OK.

The newly created publish profile appears as a selection in the Current Profile pop-up menu of the Publish Settings dialog box.

3 Specify the publish settings for your document in the Publish Settings dialog box (File > Publish Settings) and click OK. For more information about configuring publish settings, see "Publishing Flash documents" on page 281.

Duplicating a publish profile

If you've modified publish settings for a publish profile and you'd like to save the modifications, you can create a duplicate profile.

To duplicate a publish profile:

1 From the Current Profile pop-up menu in the Publish Settings dialog box (File > Publish Settings) select the publish profile that you want to copy.

2 Click the Duplicate Profile button.

3 In the Duplicate Profile dialog box, enter the profile name in the Duplicate Name text box and click OK.

The duplicate publish profile appears as a selection in the Current Profile pop-up menu of the Publish Settings dialog box.

Modifying a publish profile

To modify a publish profile, you simply change the settings in the Publish Settings dialog box.

To modify a publish profile:

1 From the Current Profile pop-up menu in the Publish Settings dialog box (File > Publish Settings), select the publish profile that you wish to copy.

2 Specify the new publish settings for your document and click OK. For details about how to select options in the dialog box, see "Publishing Flash documents" on page 281.

Exporting a publish profile

You can export a publish profile as an XML file for import into other documents. After import, the publish profile appears in the Publish Settings dialog box as an option in the Current Profile pop-up menu.

To export a publish profile:

1 From the Current Profile pop-up menu in the Publish Settings dialog box (File > Publish Settings), select the publish profile that you want to export.

2 Click the Import/Export Profile button and select Export.

3 In the Export Profile dialog box, either accept the default location in which to save the publish profile or browse to a new location and click Save.

Importing a publish profile

Other users can create and export publish profiles, which you in turn can import and select as a publish settings option.

To import a publish profile:

1 In the Publish Settings dialog box (File > Publish Settings), click Import/Export Profile, and then select Import.

2 In the Import Profile dialog box, browse to the publish profile XML file and click Open.

Deleting a publish profile

When you no longer need a publish profile, you can delete it from the document.

To delete a publish profile:

1 In the Publish Settings dialog box (File > Publish Settings), select the publish profile that you want to delete in the Current Profile pop-up menu.

2 Click the Delete Profile button. In the dialog box that requests confirmation of the deletion, click OK.

About HTML publishing templates

A Flash HTML template is a text file that contains both unchanging HTML code and template code or variables (which differ from ActionScript variables). When you publish a Flash SWF file, Flash replaces the variables in the template you selected in the Publish Settings dialog box with your HTML settings, and produces an HTML page with your SWF file embedded.

Flash includes various templates, suitable for most users' needs, that eliminate the need to edit an HTML page with the Flash SWF file. For example, one template simply places a Flash SWF file on the generated HTML page so that users can view it through a web browser if the plug-in is installed.

You can easily use the same template, change the settings, and publish a new HTML page. If you're proficient in HTML, you can also create your own templates using any HTML editor. Creating a template is the same as creating a standard HTML page, except that you replace specific values pertaining to a Flash SWF file with variables that begin with a dollar sign ($).

Flash HTML templates have the following characteristics:

- A one-line title that appears on the Template pop-up menu.
- A longer description that appears when you click the Info button.
- Template variables beginning with $ that specify where parameter values should be substituted when Flash generates the output file.

  Note: Use a backslash and dollar sign (\ $) if you need to use a $ for another purpose in the document.

- HTML `object` and `embed` tags that follow the tag requirements of Microsoft Internet Explorer and Netscape Communicator/Navigator, respectively. To display a SWF file properly on an HTML page, you must follow these tag requirements. Internet Explorer opens a Flash SWF file using the `object` HTML tag; Netscape uses the `embed` tag. For more information, see "Using object and embed tags" on page 302.

Customizing HTML publishing templates

If you're familiar with HTML, you can modify HTML template variables to create an image map, a text report or a URL report, or to insert your own values for some of the most common Flash `object` and `embed` parameters (for Internet Explorer and Netscape Communicator/Navigator, respectively).

Flash templates can include any HTML content for your application, or even code for special interpreters such as ColdFusion and ASP.

To modify an HTML publishing template:

1 Using an HTML editor, open the Flash HTML template you want to change. These templates can be found in the following locations.

For Windows operating systems:

Windows 2000 or XP <boot drive>:\Documents and Settings\<user>\Local Settings\Application Data\Macromedia\Flash MX 2004\<language>\Configuration\HTML

- The <boot drive> is the drive from which 2000 or XP boots (usually C).
- The <user> is the user name of the person logged in to the 2000 or XP.
- The <language> is set to an abbreviated language name. For example, in the US, <language> is set to "en" for English.

Note: The Application Data folder is normally a hidden folder; you may need to change your Windows Explorer settings to see this folder.

Windows 98 <boot drive>:\Program Files\Macromedia\Flash MX 2004\<language>\First Run\HTML

For Macintosh operating systems:

Macintosh OS X 10.1.5 and 10.2.6 and later <Macintosh HD>/Applications/Macromedia Flash MX 2004/First Run/HTML

2 Edit the template as needed.

- For information on variables supported in Flash, see the table that follows this procedure.
- For information on creating an image map or a text or URL report, or to insert your own values for `object` and `embed` parameters, see the sections for those topics, following this procedure.

3 When you finish editing the variables, save the template in the same folder you retrieved it from.

4 To apply the template settings to your Flash SWF file, select File > Publish Settings, select the HTML panel, and select the template you modified.

Flash changes only the template variables in the template selected in the Publish Settings dialog box.

5 Select your remaining publish settings, and click OK. For more information, see "Publishing Flash documents" on page 281.

Using HTML template variables

The following table lists the template variables that Flash recognizes. For a definition of all the tags these variables work with, see "Editing Flash HTML settings" on page 301.

Attribute/parameter	Template variable
`Template title`	`$TT`
`Template description start`	`$DS`
`Template description finish`	`$DF`
`Flash (SWF file) title`	`$TI`
`Width`	`$WI`
`Height`	`$HE`
`Movie`	`$MO`

Attribute/parameter	Template variable
`HTML alignment`	`$HA`
`Looping`	`$LO`
`Parameters for object`	`$PO`
`Parameters for embed`	`$PE`
`Play`	`$PL`
`Quality`	`$QU`
`Scale`	`$SC`
`Salign`	`$SA`
`Wmode`	`$WM`
`Devicefont`	`$DE`
`Bgcolor`	`$BG`
`Movie text (area to write movie text)`	`$MT`
`Movie URL (location of SWF file URL)`	`$MU`
`Image width (unspecified image type)`	`$IW`
`Image height (unspecified image type)`	`$IH`
`Image filename (unspecified image type)`	`$IS`
`Image map name`	`$IU`
`Image map tag location`	`$IM`
`QuickTime width`	`$QW`
`QuickTime height`	`$QH`
`QuickTime filename`	`$QN`
`GIF width`	`$GW`
`GIF height`	`$GH`
`GIF filename`	`$GN`
`JPEG width`	`$JW`
`JPEG height`	`$JH`
`JPEG filename`	`$JN`
`PNG width`	`$PW`
`PNG height`	`$PH`
`PNG filename`	`$PN`

Creating an image map

Flash can generate an image map to display any image and maintain the function of buttons that link to URLs. When an HTML template includes the `$IM` template variable, Flash inserts the image map code. The `$IU` variable identifies the name of the GIF, JPEG, or PNG file.

To create an image map:

1 In your Flash document, select the keyframe to be used for the image map and label it **#Map** in the frame Property inspector (select Window > Properties if the Property inspector is not visible). You can use any keyframe with buttons that have attached Get URL actions.

  If you don't create a frame label, Flash creates an image map using the buttons in the last frame of the SWF file. This option generates an embedded image map, not an embedded Flash SWF file.

2 To select the frame to be used for displaying the image map, do one of the following:
  - For PNG or GIF files, label the frame to be used for display as **#Static**.
  - For JPEG, during the publish operation, place the playhead on the frame to be used for display.

3 In an HTML editor, open the HTML template you'll modify. Flash stores HTML templates in the following location: <boot drive>:\Program Files\Macromedia\Flash MX 2004\<language>\First Run\HTML.

4 Save your template.

5 Select File > Publish Settings, click the Format tab, and select a format for the image map—GIF, JPEG, or PNG.

6 Click OK to save your settings.

  For example, inserting the following code in a template:

```
$IM
<img src=$IS usemap=$IU width=$IW height=$IH BORDER=0>
```

  might produce this code in the HTML document created by the Publish command:

```
<map name="mymovie">
<area coords="130,116,214,182" href="http://www.macromedia.com">
</map>
<img src="mymovie.gif" usemap="#mymovie" width=550 height=400 border=0>
```

Creating a text report

The `$MT` template variable causes Flash to insert all the text from the current Flash SWF file as a comment in the HTML code. This is useful for indexing the content of a SWF file and making it visible to search engines.

Creating a URL report

The `$MU` template variable makes Flash generate a list of the URLs referred to by actions in the current SWF file and insert it at the current location as a comment. This enables link verification tools to see and verify the links in the SWF file.

Using shorthand template variables

The `$PO` (for `object` tags) and `$PE` (for `embed` tags) template variables are useful shorthand elements. Both variables cause Flash to insert into a template any nondefault values for some of the most common Flash `object` and `embed` parameters, including `PLAY` (`$PL`), `QUALITY` (`$QU`), `SCALE` (`$SC`), `SALIGN` (`$SA`), `WMODE` (`$WM`), `DEVICEFONT` (`$DE`), and `BGCOLOR` (`$BG`). For an example of these variables, see the sample template in the following section.

Sample template

The Default.HTML template file in Flash, shown here as a sample, includes many of the commonly used template variables.

```
$TTFlash Only
$DS
Display Macromedia Flash Movie in HTML.
$DF

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
  www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
$CS
<title>$TI</title>
</head>
<body bgcolor="$BG">
<!-- url's used in the movie -->
$MU
<!-- text used in the movie -->
$MT
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" codebase="http://
  download.macromedia.com/pub/shockwave/cabs/flash/
  swflash.cab#version=$FV,0,0,0" width="$WI" height="$HE" id="$TI"
  align="$HA">
<param name="allowScriptAccess" value="sameDomain" />
$PO
<embed $PEwidth="$WI" height="$HE" name="$TI" align="$HA"
  allowScriptAccess="sameDomain" type="application/x-shockwave-flash"
  pluginspage="http://www.macromedia.com/go/getflashplayer" />
</object>
</body>
</html>
```

Editing Flash HTML settings

An HTML document is needed to play a Flash SWF file in a web browser and specify browser settings. If you are experienced with HTML, you can change or enter HTML parameters manually in an HTML editor, or create your own HTML files to control a Flash SWF file.

You can also have Flash create the HTML document automatically when you publish a SWF file; see "Publishing Flash documents" on page 281. For information on customizing HTML templates included in Flash, see "Customizing HTML publishing templates" on page 297.

Using object and embed tags

To display a Flash SWF file in a web browser, an HTML document must use the `object` and `embed` tags with the proper parameters.

For `object`, four settings (`height`, `width`, `classid`, and `codebase`) are attributes that appear within the `object` tag; all others are parameters that appear in separate, named `param` tags. For example:

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" width="100"
height="100" codebase="http://active.macromedia.com/flash7/cabs/
swflash.cab#version=7,0,0,0">
<param name="movie" value="moviename.swf">
<param name="play" value="true">
<param name="loop" value="true">
<param name="quality" value="high">
</object>
```

For the `embed` tag, all settings (such as `height`, `width`, `quality`, and `loop`) are attributes that appear between the angle brackets of the opening `embed` tag. For example:

```
<embed src="moviename.swf" width="100" height="100" play="true"
loop="true" quality="high"
pluginspage="http://www.macromedia.com/shockwave/download/
  index.cgi?P1_Prod_Version=ShockwaveFlash">
</embed>
```

To use both tags together, position the `embed` tag just before the closing `object` tag, as follows:

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" width="100"
height="100" codebase="http://active.macromedia.com/flash7/cabs/
swflash.cab#version=6,0,0,0">
<param name="movie" value="moviename.swf">
<param name="play" value="true">
<param name="loop" value="true">
<param name="quality" value="high">

<embed src="moviename.swf" width="100" height="100" play="true"
loop="true" quality="high"
pluginspage="http://www.macromedia.com/shockwave/download/
  index.cgi?P1_Prod_Version=ShockwaveFlash">
</embed>

</object>
```

Note: If you use both the `object` and the `embed` tags, use identical values for each attribute or parameter to ensure consistent playback across browsers. The parameter `swflash.cab#version=6,0,0,0` is optional, and you can omit it if you don't want to check for version number.

Parameters and attributes

The following tag attributes and parameters describe the HTML code created by the Publish command. You can refer to this list as you write your own HTML to display Flash content. Unless noted, all items apply to both `object` and `embed` tags. Optional entries are noted. Parameters are used with the `object` tag and are recognized by Internet Explorer, whereas, the `embed` tag is recognized by Netscape. Attributes are used with both the `object` tag and the `embed` tag. When customizing a template, you can substitute a template variable listed here for the value. See "Customizing HTML publishing templates" on page 297.

Note: The attributes and parameters listed in this section are shown in lowercase letters purposely to comply with the XHTML standard.

devicefont attribute/parameter

Value

`true` | *`false`*

Template variable: `$DE`

Description

(Optional) Specifies whether static text objects for which the Device Font option is not selected will be drawn using a device font anyway, if the fonts needed are available from the operating system.

src attribute

Value

`movieName``.swf`

Template variable: `$MO`

Description

Specifies the name of the SWF file to be loaded. Applies to `embed` only.

movie parameter

Value

`movieName``.swf`

Template variable: `$MO`

Description

Specifies the name of the SWF file to be loaded. Applies to `object` only.

classid attribute

Value

`clsid:d27cdb6e-ae6d-11cf-96b8-444553540000`

Description

Identifies the ActiveX control for the browser. The value must be entered exactly as shown. Applies to `object` only.

width attribute

Value

n or *n%*

Template variable: `$WI`

Description

Specifies the width of the application either in pixels or as a percentage of the browser window.

height attribute

Value

n or *n%*

Template variable: `$HE`

Description

Specifies the height of the application either in pixels or as a percentage of the browser window.

Note: Because Flash applications are scalable, their quality won't degrade at different sizes if the aspect ratio is maintained. (For example, the following sizes all have a 4:3 aspect ratio: 640 x 480 pixels, 320 x 240 pixels, and 240 x 180 pixels.)

codebase attribute

Value

`http://active.macromedia.com/flash7/cabs/swflash.cab#version=7,0,0,0`

Description

Identifies the location of the Flash Player ActiveX control so that the browser can automatically download it if it is not already installed. The value must be entered exactly as shown. Applies to `object` only.

pluginspage attribute

Value

```
http://www.macromedia.com/shockwave/download/
index.cgi?P1_Prod_Version=ShockwaveFlash
```

Description

Identifies the location of the Flash Player plug-in so that the user can download it if it is not already installed. The value must be entered exactly as shown. Applies to `embed` only.

swliveconnect attribute

Value

`true | false`

Description

(Optional) Specifies whether the browser should start Java when loading Flash Player for the first time. The default value is `false` if this attribute is omitted. If you use JavaScript and Flash on the same page, Java must be running for the `fscommand()` function to work. However, if you are using JavaScript only for browser detection or another purpose unrelated to `fscommand()`actions, you can prevent Java from starting by setting `SWLIVECONNECT` to `false`. You can also force Java to start when you are not using JavaScript with Flash by explicitly setting the `SWLIVECONNECT` attribute to `true`. Starting Java substantially increases the time it takes to start a SWF file; set this tag to `true` only when necessary. Applies to `embed` only.

Use the `fscommand()`action to start Java from a stand-alone projector file.

play attribute/parameter

Value

`true | false`

Template variable: `$PL`

Description

(Optional) Specifies whether the application begins playing immediately on loading in the browser. If your Flash application is interactive, you may want to let the user initiate play by clicking a button or performing some other task. In this case, set the `play` attribute to `false` to prevent the application from starting automatically. The default value is `true` if this attribute is omitted.

loop attribute/parameter

Value

`true` | `false`

Template variable: `$LO`

Description

(Optional) Specifies whether the Flash content repeats indefinitely or stops when it reaches the last frame. The default value is `true` if this attribute is omitted.

quality attribute/parameter

Value

`low` | `medium` | `high` | `autolow` | `autohigh` | `best`

Template variable: `$QU`

Description

(Optional) Specifies the level of anti-aliasing to be used during playback of your application. Because anti-aliasing requires a faster processor to smooth each frame of the SWF file before it is rendered on the viewer's screen, select a value based on whether speed or appearance is your top priority:

Low favors playback speed over appearance and never uses anti-aliasing.

Autolow emphasizes speed at first but improves appearance whenever possible. Playback begins with anti-aliasing turned off. If Flash Player detects that the processor can handle it, anti-aliasing is turned on.

Autohigh emphasizes playback speed and appearance equally at first but sacrifices appearance for playback speed if necessary. Playback begins with anti-aliasing turned on. If the frame rate drops below the specified frame rate, anti-aliasing is turned off to improve playback speed. Use this setting to emulate the Antialias command in Flash (View > Preview Mode > Antialias).

Medium applies some anti-aliasing and does not smooth bitmaps. It produces a better quality than the Low setting, but a lower quality than the High setting.

High favors appearance over playback speed and always applies anti-aliasing. If the SWF file does not contain animation, bitmaps are smoothed; if the SWF file has animation, bitmaps are not smoothed.

Best provides the best display quality and does not consider playback speed. All output is anti-aliased and all bitmaps are smoothed.

The default value for `quality` is `high` if this attribute is omitted.

bgcolor attribute/parameter

Value

#RRGGBB (hexadecimal RGB value)

Template variable: `$BG`

Description

(Optional) Specifies the background color of the application. Use this attribute to override the background color setting specified in the Flash SWF file. This attribute does not affect the background color of the HTML page.

scale attribute/parameter

Value

`showall` | `noborder` | `exactfit`

Template variable: `$SC`

Description

(Optional) Defines how the application is placed within the browser window when `width` and `height` values are percentages.

Showall (Default) makes the entire Flash content visible in the specified area without distortion while maintaining the original aspect ratio of the. Borders may appear on two sides of the application.

Noborder scales the Flash content to fill the specified area, without distortion but possibly with some cropping, while maintaining the original aspect ratio of the application.

Exactfit makes the entire Flash content visible in the specified area without trying to preserve the original aspect ratio. Distortion may occur.

The default value is `showall` if this attribute is omitted (and `width` and `height` values are percentages).

align attribute

Value

`Default` | `L` | `R` | `T` | `B`

Template variable: `$HA`

Description

Specifies the `align` value for the `object`, `embed`, and `img` tags and determines how the Flash SWF file is positioned within the browser window.

Default centers the application in the browser window and crops edges if the browser window is smaller than the application.

L, R, T, and **B** align the application along the left, right, top, and bottom edge, respectively, of the browser window and crop the remaining three sides as needed.

salign parameter

Value

`L` | `R` | `T` | `B` | `TL` | `TR` | `BL` | `BR`

Template variable: `$SA`

Description

(Optional) Specifies where a scaled Flash SWF file is positioned within the area defined by the `width` and `height` settings. For more information about these conditions, see "scale attribute/parameter" on page 307.

L, **R**, **T**, and **B** align the application along the left, right, top or bottom edge, respectively, of the browser window and crop the remaining three sides as needed.

TL and **TR** align the application to the top left and top right corner, respectively, of the browser window and crop the bottom and remaining right or left side as needed.

BL and **BR** align the application to the bottom left and bottom right corner, respectively, of the browser window and crop the top and remaining right or left side as needed.

If this attribute is omitted, the Flash content is centered in the browser window.

base attribute

Value

base directory or URL

Description

(Optional) Specifies the base directory or URL used to resolve all relative path statements in the Flash SWF file. This attribute is helpful when your SWF files are kept in a different directory from your other files.

menu attribute/parameter

Value

`true` | `false`

Template variable: `$MF`

Description

(Optional) Specifies what type of menu is displayed when the viewer right-clicks (Windows) or Command-clicks (Macintosh) the application area in the browser.

true displays the full menu, allowing the user a variety of options to enhance or control playback.

false displays a menu that contains only the About Macromedia Flash Player 6 option and the Settings option.

The default value is `true` if this attribute is omitted.

wmode attribute/parameter

Value

`Window | Opaque | Transparent`

Template variable: `$WM`

Description

(Optional) Lets you take advantage of the transparent Flash content, absolute positioning, and layering capabilities available in Internet Explorer 4.0. This attribute/parameter works only in Windows with the Flash Player ActiveX control.

Window plays the application in its own rectangular window on a web page. Window indicates that the Flash application has no interaction at all with HTML layers and is always the topmost item.

Opaque makes the application hide everything behind it on the page.

Transparent makes the background of the HTML page show through all the transparent portions of the application, and may slow animation performance.

Opaque windowless and **Transparent windowless** both interact with HTML layers, allowing layers above the SWF file to block out the application. The difference between the two is that Transparent allows transparency so that HTML layers below the SWF file might show through if a section of the SWF file has transparency.

The default value is `Window` if this attribute is omitted. Applies to `object` only.

allowscriptaccess attribute/parameter

Value

`always | never | samedomain`

Description

Use `allowscriptaccess` to enable your Flash application to communicate with the HTML page hosting it. This is required because `fscommand()` and `getURL()` operations can cause JavaScript to use the permissions of the HTML page, which may be different from the permissions of your Flash application. This has important implications for cross-domain security.

always permits scripting operations at all times.

never forbids all scripting operations.

samedomain permits scripting operations only if the Flash application is from the same domain as the HTML page.

The default value used by all HTML publish templates is `samedomain`.

Previewing the publishing format and settings

To preview your Flash SWF file with the publishing format and settings you've selected, you can use the Publish Preview command. This command exports the file and opens the preview in the default browser. If you preview a QuickTime movie, Publish Preview starts the QuickTime Movie Player. If you preview a projector, Flash starts the projector.

To preview a file with the Publish Preview command:

1 Define the file's export options using the Publish Settings command; see "Publishing Flash documents" on page 281.

2 Select File > Publish Preview, and from the submenu select the file format you want to preview.

Using the current Publish Settings values, Flash creates a file of the specified type(s) in the same location as the Flash document (FLA). This file remains in this location until you overwrite or delete it.

Using Flash Player

Flash Player plays Flash content in the same way as it appears in a web browser or an ActiveX host application. The player is installed along with Flash application. When you double-click Flash content, the operating system starts Flash Player, which in turn plays the SWF file. You can use the player to make Flash content viewable for users who aren't using a web browser or an ActiveX host application.

You can control Flash content in Flash Player using menu commands and the `fscommand()` function. For example, to make Flash Player take over the entire screen, you assign `fscommand()` to a frame or button and then select the `fullscreen` command with the `true` parameter.

You can also print Flash content frames using the Flash Player context menu. See "Printing from the Flash Player context menu" on page 347.

To control applications from the Flash Player, do one of the following:

- Open a new or existing file by selecting File > New or File > Open.
- Change your view of the application by selecting View > Magnification, and from the submenu select Show All, Zoom In, Zoom Out, or 100%.
- Control Flash content playback by selecting Control > Play, Rewind, or Loop.

About configuring a web server for Flash

When your files are accessed from a web server, the server must properly identify them as Flash content in order to display them. If the MIME type is missing or not properly delivered by the server, the browser may display error messages or a blank window with a puzzle piece icon.

If your server is not properly configured, you (or your server's administrator) must add the Flash SWF file MIME types to the server's configuration files and associate the following MIME types with the SWF file extensions:

- MIME type application/x-shockwave-flash has the .swf file extension.
- MIME type application/futuresplash has the .spl file extension.

If you are administering your own server, consult your server software documentation for instructions on adding or configuring MIME types. If you are not administering your own server, contact your Internet service provider, webmaster, or server administrator to add the MIME type information.

If your site is on a Macintosh server, you must also set the following parameters: Action: Binary; Type: SWFL; and Creator: SWF2.

CHAPTER 16 Exporting

The Export Movie command in Macromedia Flash MX 2004 and Macromedia Flash MX Professional 2004 lets you create content that can be edited in other applications and export Flash content directly into a single format. For example, you can export an entire document as a Flash SWF file; as a series of bitmap images; as a single frame or image file; or as moving and still images in various formats, including GIF, JPEG, PNG, BMP, PICT, QuickTime, or AVI.

When you export a Flash file in the SWF format, text is encoded as Unicode, providing support for international character sets, including double-byte fonts. Macromedia Flash Player 6 and later versions support Unicode encoding. For more information, see Chapter 13, "Creating Multilanguage Text," on page 215.

If you have Macromedia Dreamweaver, you can add Flash content to your website easily. Dreamweaver generates all the needed HTML code. You can start Flash from within Dreamweaver to update the Flash content. See "Updating Flash content for Dreamweaver UltraDev" on page 318.

Exporting Flash content and images

To prepare Flash content for use in other applications or to export the contents of the current Flash document in a particular file format, you use the Export Movie and Export Image commands. The Export commands do not store export settings separately with each file, as does the Publish command. (You use the Publish command to create all the files you need to put Flash content on the web. See "Publishing Flash documents" on page 281.)

The Export Movie command lets you export a Flash document to a still-image format and create a numbered image file for every frame in the document. You can also use Export Movie to export the sound in a document to a WAV file (Windows only).

To export the contents of the current frame or the currently selected image to one of the still-image formats, or to a single-frame Flash Player application, you use the Export Image command.

Keep the following considerations in mind:

- When you export a Flash image as a vector-graphic file (in Adobe Illustrator format), you preserve its vector information. You can edit these files in other vector-based drawing programs, but you can't import these images into most page-layout and word-processing programs.
- When you save a Flash image as a bitmap GIF, JPEG, PICT (Macintosh), or BMP (Windows) file, the image loses its vector information and is saved with pixel information only. You can edit Flash images exported as bitmaps in image editors such as Adobe Photoshop, but you can no longer edit them in vector-based drawing programs.

To export a Flash document or image:

1. Open the Flash document you want to export, or if you are exporting an image from the document, select the frame or image in the current document that you want to export.
2. Select File > Export Movie or File > Export Image.
3. Enter a name for the output file.
4. Select the file format from the Format pop-up menu.
5. Click Save.

   If the format you selected requires more information, an Export dialog box appears.
6. Set the export options for the format you selected. See the following section.
7. Click OK, then click Save.

About export file formats

You can export Flash content and images in more than a dozen different formats, as shown in the table that follows. Flash content is exported as sequences, and images are exported as individual files. PNG is the only cross-platform bitmap format that supports transparency (as an alpha channel). Some nonbitmap export formats do not support alpha (transparency) effects or mask layers.

For more information on a specific file format, see the sections that follow.

File type	Extension	Windows	Macintosh
"Adobe Illustrator" on page 313	.ai	✓	✓
"Animated GIF, GIF Sequence, and GIF Image" on page 313	.gif	✓	✓
"Bitmap (BMP)" on page 314	.bmp	✓	
"DXF Sequence and AutoCAD DXF Image" on page 314	.dxf	✓	✓
"Enhanced Metafile (Windows)" on page 314	.emf	✓	
"EPS 3.0 with Preview" on page 314	.eps	✓	✓
"Flash document (SWF)" on page 314	.swf	✓	✓
"Macromedia Flash Video (FLV)" on page 315	.flv	✓	✓
"JPEG Sequence and JPEG Image" on page 315	.jpg	✓	✓

File type	Extension	Windows	Macintosh
"PICT (Macintosh)" on page 315	.pct		✔
"PNG Sequence and PNG Image" on page 316	.png	✔	✔
"QuickTime" on page 316	.mov	✔	✔
"QuickTime Video (Macintosh)" on page 316	.mov		✔
"WAV audio (Windows)" on page 317	.wav	✔	
"Windows AVI (Windows)" on page 317	.avi	✔	
"Windows Metafile" on page 317	.wmf	✔	

Adobe Illustrator

The Adobe Illustrator format is ideal for exchanging drawings between Flash and other drawing applications such as Macromedia FreeHand. This format supports very accurate conversion of curve, line style, and fill information. Flash supports import and export of the Adobe Illustrator 88, 3, 5, 6, and 8 through 10 formats. (See "Importing Adobe Illustrator, EPS, or PDF files" on page 125.) Flash does not support the Photoshop EPS format or EPS files generated using the Print command.

Versions of the Adobe Illustrator format before 5 do not support gradient fills, and only version 6 supports bitmaps.

The Export Adobe Illustrator dialog box lets you select the Adobe Illustrator version—88, 3.0, 5.0, or 6.0.

You can use the Macromedia Flashwriter plug-in to export files in SWF format from Adobe Illustrator 8. Adobe Illustrator versions 9 and 10 have built-in support for SWF export, so the Macromedia Flashwriter plug-in is not needed with those applications.

Animated GIF, GIF Sequence, and GIF Image

The Animated GIF, GIF Sequence, and GIF Image option lets you export files in the GIF format. The settings are the same as those available on the GIF tab in the Publish Settings dialog box, with the following exceptions:

Resolution is set in dots per inch (dpi). You can enter a resolution or click Match Screen to use the screen resolution.

Include lets you select to export the minimum image area or specify the full document size.

Colors lets you set the number of colors that can be used to create the exported image—black-and-white; 4-, 6-, 16-, 32-, 64-, 128-, or 256-color; or Standard Color (the standard 216-color, browser-safe palette).

You can also choose to interlace, smooth, make transparent, or dither solid colors. For information on these options, see "Configuring publish settings for Flash Player detection" on page 287.

Animation is available for the Animated GIF export format only and lets you enter the number of repetitions, where 0 repeats endlessly.

Bitmap (BMP)

The Bitmap (BMP) format lets you create bitmap images for use in other applications. The Bitmap Export Options dialog box has these options:

Dimensions sets the size of the exported bitmap image in pixels. Flash ensures that the size you specify always has the same aspect ratio as your original image.

Resolution sets the resolution of the exported bitmap image in dots per inch (dpi) and has Flash automatically calculate width and height based on the size of your drawing. To set the resolution to match your monitor, select Match Screen.

Color Depth specifies the bit depth of the image. Some Windows applications do not support the newer 32-bit depth for bitmap images; if you have problems using a 32-bit format, use the older 24-bit format.

Smooth applies anti-aliasing to the exported bitmap. Anti-aliasing produces a higher-quality bitmap image, but it may create a halo of gray pixels around an image placed on a colored background. Deselect this option if a halo appears.

DXF Sequence and AutoCAD DXF Image

The DXF Sequence and AutoCad DXF Image 3D format lets you export Flash content as AutoCAD DXF release 10 files, so that they can be brought into a DXF-compatible application for additional editing.

This format has no definable export options.

Enhanced Metafile (Windows)

Enhanced Metafile Format (EMF) is a graphics format available in Windows 95 and Windows NT that saves both vector and bitmap information. EMF supports the curves used in Flash drawings better than the older Windows Metafile format. However, some applications do not yet support this graphics format.

This format has no definable export options.

EPS 3.0 with Preview

You can export the current frame as an EPS 3 file for placement in another application, such as a page layout application. An EPS (encapsulated PostScript) file can be printed by a PostScript printer. As an option, you can include a bitmap preview with the exported EPS file for applications that can import and print the EPS files (such as Microsoft Word and Adobe PageMaker) but that can't display them onscreen.

Flash has no definable exporting options for EPS files.

Flash document (SWF)

You can export the entire document as a Flash SWF file, to place the Flash content in another application, such as Dreamweaver. You can select the same options for exporting a document as you can for publishing the document. See "Publishing Flash documents" on page 281.

Macromedia Flash Video (FLV)

The Macromedia Flash Video (FLV) file format lets you import or export a static video stream with encoded audio. This format is intended for use with communications applications, such as video conferencing and files that contain screen share encoded data exported from the Flash Communication Server.

When you export video clips with streaming audio in FLV format, the audio is compressed using the Streaming Audio settings in the Publish Settings dialog box. For information on audio settings, see "Setting publish options for the Flash SWF file format" on page 282.

Files in the FLV format are compressed with the Sorensen codec. See "About the Sorenson Spark codec" on page 165.

To export a video clip in FLV format:

1 Select the video clip in the Library panel.

2 Select Properties from the Library options menu.

3 In the Embedded Video Properties dialog box, click Export.

4 In the Save As dialog box, enter a name for the exported file. Select a location where it will be saved, and click Save.

5 In the Embedded Video Properties dialog box, click OK to close the dialog box.

JPEG Sequence and JPEG Image

The JPEG export options match the JPEG Publish Settings options with one exception: the Match Screen export option makes the exported image match the size of the Flash content as it appears on your screen. The Match Movie publishing option makes the JPEG image the same size as the Flash content and maintains the aspect ratio of the original image.

For more information, see "Specifying publish settings for JPEG files" on page 290.

PICT (Macintosh)

PICT is the standard graphics format on the Macintosh and can contain bitmap or vector information. Use the Export PICT dialog box to set the following options:

Dimensions sets the size of the exported bitmap image specified in pixels. Flash ensures that the size you specify always has the same aspect ratio as your original image.

Resolution sets the resolution in dots per inch (dpi) and has Flash automatically calculate width and height based on the size of your drawing. To set the resolution to match your monitor, select Match Screen. Bitmap PICT images usually look best onscreen with 72-dpi resolution.

Include sets the portion of the document to be exported, either Minimum Image Area or Full Document Size.

Color Depth designates whether the PICT file is object-based or bitmap. Object-based images generally look better when printed, and scaling doesn't affect their appearance. Bitmap PICT images normally look best displayed onscreen and can be manipulated in applications such as Adobe Photoshop. You can also select a variety of color depths with bitmap PICT files.

Include Postscript is available only for an object-based PICT file to include information that optimizes printing on a PostScript printer. This information makes the file larger and may not be recognized by all applications.

Smooth Bitmap is available only for a bitmap PICT. This option applies anti-aliasing in order to smooth jagged edges of a bitmap image.

PNG Sequence and PNG Image

The PNG export settings options are similar to the PNG publish settings options (see "Specifying publish settings for PNG files" on page 291), with the following exceptions:

Dimensions sets the size of the exported bitmap image to the number of pixels you enter in the Width and Height fields.

Resolution lets you enter a resolution in dots per inch (dpi). To use the screen resolution and maintain the aspect ratio of your original image, select Match Screen.

Colors is the same as the Bit Depth option in the PNG Publish Settings tab and sets the number of bits per pixel to use in creating the image. For a 256-color image, select 8-bit; for thousands of colors, select 24-bit; for thousands of colors with transparency (32 bits) select 24-bit with Alpha. The higher the bit depth, the larger the file.

Include lets you choose to export the minimum image area or specify the full document size.

Filter options match those in the PNG Publish Settings tab.

When exporting a PNG sequence or PNG image, you can also apply other options in the PNG Publish Settings, such as Interlace, Smooth, and Dither Solid Colors.

QuickTime

The QuickTime export option creates an application with a Flash track in the same QuickTime format that is installed on your computer. This export format lets you combine the interactive features of Flash with the multimedia and video features of QuickTime in a single QuickTime 4 movie, which can be viewed by anyone with the QuickTime 4 plug-in.

If you import a video clip (in any format) into a document as an embedded file, you can publish the document as a QuickTime movie. If you have imported a video clip in QuickTime format into a document as a linked file, you can also publish the document as a QuickTime movie.

When you export Flash content as a QuickTime movie, all layers in the Flash document are exported as a single Flash track, unless the Flash document contains an imported QuickTime movie. The imported QuickTime movie remains in QuickTime format in the exported application.

These export options are identical to QuickTime publish options. See "Specifying publish settings for QuickTime movies" on page 293.

QuickTime Video (Macintosh)

The QuickTime Video format converts the Flash document into a sequence of bitmaps embedded in the file's video track. The Flash content is exported as a bitmap image without any interactivity. This format is useful for editing Flash content in a video-editing application.

The Export QuickTime Video dialog box contains the following options:

Dimensions specifies a width and height in pixels for the frames of a QuickTime movie. By default, you can specify only the width or the height, and the other dimension is automatically set to maintain the aspect ratio of your original document.To set both the width and the height, deselect Maintain Aspect Ratio.

Format selects a color depth. Options are black-and-white; 4-, 8-, 16-, or 24-bit color; and 32-bit color with alpha (transparency).

Smooth applies anti-aliasing to the exported QuickTime movie. Anti-aliasing produces a higher-quality bitmap image, but it may cause a halo of gray pixels to appear around images when placed over a colored background. Deselect the option if a halo appears.

Compressor selects a standard QuickTime compressor. For more information, see your QuickTime documentation.

Quality controls the amount of compression applied to your Flash content. The effect depends on the compressor selected.

Sound Format sets the export rate for sounds in the document. Higher rates yield better fidelity and larger files. Lower rates save space.

WAV audio (Windows)

The WAV Export Movie option exports only the sound file of the current document to a single WAV file. You can specify the sound format of the new file.

Select Sound Format to determine the sampling frequency, bit rate, and stereo or mono setting of the exported sound. Select Ignore Event Sounds to exclude events sounds from the exported file.

Windows AVI (Windows)

This format exports a document as a Windows video but discards any interactivity. The standard Windows movie format, Windows AVI, is a good format for opening a Flash animation in a video-editing application. Because AVI is a bitmap-based format, documents that contain long or high-resolution animations can quickly become very large.

The Export Windows AVI dialog box has the following options:

Dimensions specifies a width and height, in pixels, for the frames of an AVI movie. Specify only the width or the height; the other dimension is automatically set to maintain the aspect ratio of your original document. Deselect Maintain Aspect Ratio to set both the width and the height.

Video Format selects a color depth. Some applications do not yet support the Windows 32-bit image format. If you have problems using this format, use the older 24-bit format.

Compress Video displays a dialog box for choosing standard AVI compression options.

Smooth applies anti-aliasing to the exported AVI movie. Anti-aliasing produces a higher-quality bitmap image, but it may cause a halo of gray pixels to appear around images when placed over a colored background. Deselect the option if a halo appears.

Sound Format lets you set the sample rate and size of the sound track, and whether it will be exported in mono or stereo. The smaller the sample rate and size, the smaller the exported file, with a possible trade-off in sound quality. For more information on exporting sound to the AVI format, see "Compressing sounds for export" on page 192.

Windows Metafile

Windows Metafile format is the standard Windows graphics format and is supported by most Windows applications. This format yields good results for importing and exporting files. It has no definable export options. See "Enhanced Metafile (Windows)" on page 314.

Updating Flash content for Dreamweaver UltraDev

If you have Dreamweaver UltraDev installed on your system, you can export Flash SWF files directly to a Dreamweaver UltraDev site. For more information on working with Dreamweaver UltraDev, see *Using Dreamweaver*.

In Dreamweaver UltraDev, you can add the Flash content to your page. With a single click, you can update the Flash document (FLA) and re-export the updated Flash content to UltraDev automatically.

To update Flash content for Dreamweaver UltraDev:

1. In Dreamweaver UltraDev, open the HTML page that contains the Flash content.
2. Do one of the following:
   - Select the Flash content and click Edit in the Property inspector.
   - In Design view, press Control (Windows) or Command (Macintosh) and double-click the Flash content.
   - In Design view, right-click (Windows) or Control-click (Macintosh) the Flash content and select Edit with Flash from the context menu.
   - In the Site panel, right-click (Windows) or Control-click (Macintosh) the Flash content in Design view and select Open with Flash from the context menu.

   The Flash application is started on your system.
3. If the Flash document (FLA) for the exported file does not open, a file locator dialog box appears. Navigate to the FLA file in the Open File dialog box and click Open.
4. If the user has used the Change Link Sitewide feature in Dreamweaver UltraDev, a warning is shown. Click OK to apply link changes to the Flash content. Click Don't Warn Me Again to prevent the warning message from appearing when you update the Flash content.
5. Update the Flash document (FLA) as needed in Flash.
6. To save the Flash document (FLA) and re-export the Flash content to Dreamweaver, do one of the following:
   - To update the file and close Flash, click the Done button above the upper left corner of the Stage.
   - To update the file and keep Flash open, select File > Update for Dreamweaver.

CHAPTER 17
Creating Accessible Content

You can create Flash content that is accessible to all users, including those with disabilities, using the accessibility features provided with Macromedia Flash MX 2004 and Macromedia Flash MX Professional 2004. As you design accessible Flash applications, consider how your users will interact with the content. Visually impaired users, for example, might rely on assistive technology, such as screen readers that provide an audio version of screen content, while hearing-impaired users might read text and captions in the document. Other considerations arise for users with mobility or cognitive impairments.

For a quick lesson in how to create accessible content, select Help > How Do I > Quick Tasks > Create Accessible Flash Content. To view a sample of an accessible application, see "Using the accessibility features in Flash" on page 382.

You can create accessible content with Flash by using accessibility features included in the authoring environment user interface, by taking advantage of ActionScript designed to implement accessibility, and by following recommended design and development practices. The list of recommended practices that follows is not exhaustive, but rather suggests common issues to consider. Depending on your audience needs, additional requirements might arise.

Visually impaired users For visually impaired users, including those with color blindness, keep in mind the following design recommendations:

- Use the Accessibility panel or ActionScript to provide a description of your document and nontext elements for use with a screen reader. See "Using Flash to enter accessibility information for screen readers" on page 323 and "Creating accessibility with ActionScript" on page 334.
- Describe the layout of your movie and the individual controls used to navigate through the Flash application. See "Using Flash to enter accessibility information for screen readers" on page 323.
- Design and implement a logical tab order using either the Accessibility panel or ActionScript. See "Creating a tab order index for keyboard navigation in the Accessibility panel (Flash Professional only)" on page 331 and "Using ActionScript to create a tab order for accessible objects" on page 335.
- Design the document so that constant changes in the Flash content do not unnecessarily cause screen readers to refresh. For example, you should group or hide looping elements. See "Hiding an object from the screen reader" on page 327.

- Provide captions for narrative audio. Be aware of audio in your document that might interfere with a user being able to listen to the screen reader. See "Testing accessible content" on page 336.
- Ensure that color is not the only means of conveying information. In addition, make sure that foreground and background contrast sufficiently to make text readable for people with low vision and color blindness.

Users with visual or mobility impairment For users with either visual or mobility impairment, ensure that controls are device independent (or accessible by keyboard).

Hearing-impaired users For hearing impaired users, you can caption audio content. See "Accessibility for hearing-impaired users" on page 336.

Users with cognitive impairment Users with cognitive impairments often respond best to uncluttered design that is easily navigable.

Worldwide accessibility standards

Many countries, including the United States, Australia, Canada, Japan, and countries in the European Union, have adopted accessibility standards based on those developed by the World Wide Web Consortium (W3C). The W3 publishes the Web Content Accessibility Guidelines, a document that prioritizes actions designers should take to make web content accessible. For information about the Web Accessibility Initiative, see the W3C website at www.w3.org/WAI.

In the United States, the law that governs accessibility is commonly known as Section 508, which is an amendment to the U.S. Rehabilitation Act. Section 508 prohibits federal agencies from buying, developing, maintaining, or using electronic technology that is not accessible to those with disabilities. In addition to mandating standards, Section 508 allows government employees and the public to sue agencies in federal court for noncompliance.

For additional information about Section 508, see the following websites:

- The US government-sponsored website at www.section508.gov
- The Macromedia accessibility site at www.macromedia.com/macromedia/accessibility/

Macromedia Flash Accessibility web page

For the latest information on creating and viewing accessible Flash content, including supported platforms, screen reader compatibility, articles, and accessible examples, consult the Macromedia Flash Accessibility web page at www.macromedia.com/software/Flash/productinfo/accessibility/.

Understanding screen reader technology

Screen readers are software designed to navigate through a website and read the web content aloud. Visually impaired users often rely on this technology. You can create Flash content designed for use with screen readers on only Windows platforms. Those viewing your Flash content must have Flash Player 6 or later, and Internet Explorer on Windows 98 or later.

JAWS, from Freedom Scientific, is one example of screen reader software. You can access the JAWS page of the Freedom Scientific website at www.hj.com/fs_products/software_jaws.asp. Another commonly used screen reader program is Window-Eyes, from GW Micro. To access the latest information on Window-Eyes, visit the GW Micro website at www.gwmicro.com. To enable a screen reader to read nontextual objects in your application, such as vector art and animations, you can use the Accessibility panel to associate a name and description with the object, which the screen reader reads aloud.

Screen readers help users understand what is contained in a web page or Flash document. Based on the keyboard shortcuts you define, you can allow users to navigate through your document using the screen reader with ease. See "Creating a keyboard shortcut" on page 328.

To expose graphic objects, you can provide a description using the Accessibility panel or ActionScript. See "Using Flash to enter accessibility information for screen readers" on page 323.

Because different screen reader applications use varying methods to translate information into speech, your content will vary in how it's presented to each user. As you design accessible applications, keep in mind that you have no control over how a screen reader will behave. You can only mark up the content in your Flash applications in such a way as to expose the text and ensure screen reader users can activate the controls. You only have control over the content, not the screen readers. This means that you can decide which objects in the movie are exposed to screen readers, provide descriptions for them, and decide the order in which they are exposed to screen readers. However, you cannot force screen readers to read specific text at specific times or control the manner in which that content is read. It is very important, therefore, to test your applications with a variety of screen readers to ensure that they perform as you expect. See "Testing accessible content" on page 336.

Flash and Microsoft Active Accessibility (Windows only)

Flash Player is optimized for Microsoft Active Accessibility (MSAA) which provides a highly descriptive and standardized way for applications and screen readers to communicate. MSAA is available on Windows operating systems only. For more information on Microsoft Accessibility Technology, visit the Microsoft Accessibility website at www.microsoft.com/enable/default.aspx.

The Windows ActiveX (Internet Explorer plug-in) version of Flash Player 6 supports MSAA, but the Windows Netscape and Windows stand-alone players do not.

Caution: MSAA is currently *not* supported in the opaque windowless and transparent windowless modes. (These modes are options in the HTML Publish Settings panel, available for use with the Windows version of Internet Explorer 4.0 or later, with the Flash ActiveX control.) If you need your Flash content to be accessible to screen readers, avoid using these modes.

Flash Player makes information about the following types of accessibility objects available to screen readers using MSAA. To understand how to enter accessible information for each object, see "Using Flash to enter accessibility information for screen readers" on page 323.

Dynamic or static text The principal property of a text object is its name. To comply with MSAA convention, the name is equal to the contents of the text string. A text object may also have an associated description string. Flash uses the static or dynamic text immediately above or to the left of an input text field as a label for that field.

Note: Any text that is a label is *not* passed to a screen reader. Instead, the content of that text is used as the name of the object that it labels. Labels are never assigned to buttons or text fields that have author-supplied names.

Input text fields An input text object has a value, an optional name, a description string, and a keyboard shortcut string. Like dynamic text, an input text object's name may come from a text object that is above or to the left of it.

Buttons A button object has a state (pressed or not pressed), supports a programmatic default action that causes the button to depress momentarily, and may optionally have a name, a description string, and a keyboard shortcut string. As with text input fields, for buttons, Flash uses any text entirely inside a button as a label for that button.

Note: For accessibility purposes, movie clips used as buttons with button event handlers such as `onPress`, are considered buttons, not movie clips, by Flash Player.

Components Flash UI components provide special accessibility implementation. For more information, see "Using accessible components" on page 333 and "Creating accessibility with ActionScript" on page 334.

Movie clips Movie clips are exposed to screen readers as graphic objects when they do not contain any other accessible objects, or when the Accessibility panel is used to provide a name or a description for a movie clip. When a movie clip contains other accessibly objects, the clip itself is ignored, and the objects inside it are made available to screen readers.

Note: All Flash Video objects are treated as simple movie clips.

Basic accessibility support in Flash Player

Flash Player provides some basic accessibility support for all Flash documents, whether or not they are designed using the accessibility features found in the Flash authoring tool. This generic support for documents that do not use any accessibility features includes the following:

Dynamic or static text Text is transferred to the screen reader program as a name, but with no description.

Input text Text is transferred to the screen reader. No names are transferred, except where labeling relationships are found, and no descriptions or keyboard shortcut strings are transferred.

Buttons The state of the button is transferred to the screen reader. No names are transferred, except where labeling relationships are found, and no descriptions or keyboard shortcut strings are transferred.

Documents The document state is transferred to the screen reader, but with no name or description.

Using Flash to enter accessibility information for screen readers

Screen readers read aloud a description of the content, read text, and assist users as they navigate through the user interfaces of traditional applications such as menus, toolbars, dialog boxes, and input text fields.

By default, the following objects are defined as accessible in all Flash documents and are included in the information that Flash Player provides to screen reader software:

- Dynamic text
- Input text fields
- Buttons
- Movie clips
- Entire Flash applications

Flash Player automatically provides names for static and dynamic text objects, which are simply the contents of the text. For each of the five kinds of accessible objects shown above, you can set descriptive properties for screen readers to read aloud. You can also control how Flash Player decides which objects to expose to screen readers—for example, you can specify that certain accessible objects are not exposed to screen readers at all.

The Flash Accessibility panel

One way to provide accessibility information to screen readers is to use the Flash Accessibility panel. The alternate approach is to enter accessibility information using ActionScript. See "Creating accessibility with ActionScript" on page 334.

The Accessibility panel is a self-contained property inspector that lets you set accessibility options for individual Flash objects or entire Flash applications.

If you select an object on the Stage, you can make that object accessible and then specify options such as a name, description, keyboard shortcut, and tab index order (Flash Professional only) for the object. For movie clips, you can specify whether child object information is passed to the screen reader (this option is selected by default when you make an object accessible).

With no objects selected on the Stage, you use the Accessibility panel to assign accessibility options for an entire Flash application. You can make the entire application accessible, make child objects accessible, have Flash label objects automatically, and give specific names and descriptions to objects

All objects in Flash documents must have instance names in order for you to apply accessibility options to them. You create instance names for objects in the Property inspector. The instance name is used to refer to the object in ActionScript.

To open the Accessibility panel:

1 Select Window > Other Panels > Accessibility.

2 Select from the available options:

Make Object Accessible instructs Flash Player to pass the accessibility information for an object to a screen reader. This option is selected by default; when the option is disabled, accessibility information for the object is not passed to screen readers. You might find it useful to disable this option as you test content for accessibility because some objects may be extraneous or decorative and making them accessible could produce confusing results in the Screen Reader. You can then apply a name manually to the labeled object, and hide the labeling text by unselecting Make Object Accessible. When Make Object Accessible is disabled, all other controls on the Accessibility panel are disabled.

Make Child Objects Accessible instructs Flash Player passes child object information to the screen reader. This option is for movie clips only and is selected by default. When it is enabled, Disabling this option for a movie clip causes that movie clip to appear as a simple clip in the accessible object tree, even if the clip contains text, buttons, and other objects. All objects within the movie clip are then hidden from the object tree. Like the Make Object Accessible option, this option is useful mainly for hiding extraneous objects from screen readers.

Note: If a movie clip is used as a button, meaning that it has a button event handler assigned to it, such as `onPress` or `onRelease`, the Make Child Objects Accessible option is ignored because buttons are always treated as simple clips, and their children are never examined, except in the case of labels.

Auto Label instructs Flash to automatically label objects on the Stage with the text associated with them. See "You can use the Accessibility panel to assign names to buttons and input text fields so that they are identified appropriately by the screen reader. There are two ways of doing this:" on page 325.

Name specifies the object name. Screen readers identify objects by reading these names aloud. When accessible objects don't have specified names, a screen reader might read a generic word, such as *Button*, which can be confusing.

Caution: Do not confuse object names specified in the Accessibility panel with instance names specified in the Property inspector.

Description lets you enter a description of the object to the screen reader. This description is read by the screen reader.

Shortcut is used to describe keyboard shortcuts to the users. The text entered in this text box is read by the screen reader. Entering keyboard shortcut text here does not create a keyboard shortcut for the selected object. You must provide ActionScript keyboard handlers in order to create shortcut keys. For more information, see "Creating a keyboard shortcut" on page 328.

Tab Index (Flash Professional only) creates a tab order in which objects are accessed when the user presses the tab key. The tab index feature works for keyboard navigation through a page, but not for screen reader reading order. For information on how to use this field, see "Creating a tab order index for keyboard navigation in the Accessibility panel (Flash Professional only)" on page 331.

For more information, see the Macromedia Flash Accessibility web page at www.macromedia.com/software/Flash/productinfo/accessibility/.

Choosing names for buttons, text fields, and entire Flash applications

You can use the Accessibility panel to assign names to buttons and input text fields so that they are identified appropriately by the screen reader. There are two ways of doing this:

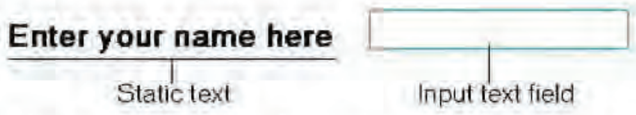
- Use the auto label feature to assign text adjacent or within the object as a label.
- Enter a specific label in the Accessibility panel name field.

Using automatic labeling

Flash automatically gives an appropriate name to a button or input text field in your document, as a text label that you have placed on top of, inside, or near a button or another text field. Labels for buttons must appear within the bounding shape of the button. For the button in the following illustration, most screen readers would first read the word *button*, then read the text label *Home*. The user could press Return or Enter to activate the button.

A form might include an input text field where users enter their names. A static text field, with the text *Name* appears next to the input text field. When Flash Player discovers an arrangement like this, it assumes that the static text object is a serving as a label for the input text field.

For example, when the following part of a form is encountered, a screen reader reads "Enter your name here."



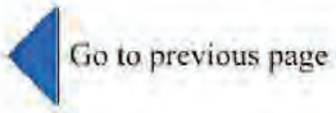
In the Accessibility panel, you can turn off automatic labeling if it is not appropriate for your document. You can also turn off automatic labeling for specific objects within your document. See "Turning off automatic labeling for an object and specifying a name" on page 327.

Providing a name for an object

If you do not want to use automatic labeling for the entire application, you can turn it off and provide names for the objects in the Accessibility panel. If you have automatic labeling turned on, you can also select specific objects and provide names for the objects in the Name text box in the Accessibility panel so that the name is used instead of the object text label.

When a button or input text field doesn't have a text label, or when the label is in a location that Flash Player can't detect, you can specify a name for the button or text field. You can also specify a name if the text label is near a button or text field, but you don't want that text to be used as that object's name.

For example, in the following graphic, the text that describes the button appears outside and to the right of the button. In this location, Flash Player does not detect the text and it is not read by the screen reader.



To rectify this, open the Accessibility panel and select the button and enter the desired name (like "left arrow") and description (like "Go to previous page") in the Name and Description text boxes, respectively. To prevent repetition, make the text object inaccessible.

Note: An object's accessibility name is unrelated to the ActionScript instance name or ActionScript variable name associated with the object. For information on how ActionScript handles instance names and variable names in text fields, see About text field instance and variable names in ActionScript Reference Guide Help. (This information generally applies to all objects.)

To specify a name and description for a button, text field, or entire Flash application:

1 Do one of the following:
- To provide a name for a button or text field, select the object on the Stage.
- To provide a name for an entire Flash application, deselect all objects on the Stage.

2 Do one of the following:
- Select Window > Property inspector if the inspector is not visible. In the Property inspector, click the Accessibility button.
- Select Window > Other Panels > Accessibility.

3 In the Accessibility panel, make sure that the Make Object Accessible (for buttons or text fields) or Make Movie Accessible (for entire Flash applications) option is selected (the default setting).

4 Enter a name for the button, text field, or Flash application in the Name text box.

5 Enter a description for the button, text field, or Flash application in the Description text box.

To define accessibility for a selected object in a movie:

1 Select the object on the Stage and do one of the following:
- Select Window > Property inspector if the inspector is not visible. In the Property inspector, click the Accessibility button.
- Select Window > Other Panels > Accessibility.

2 In the Accessibility panel, do one of the following:
- Select Make Object Accessible (the default setting) to expose the object to screen readers, and to enable other options in the panel.
- Deselect Make Object Accessible to hide the object from screen readers. This disables the other options in the panel.

3 Enter information for the selected object as needed:

Dynamic text Enter a name for the text object in the Name text box, and an optional description of the text in the Description text box. (To provide a description for static text, you must convert it to dynamic text.)

Input text fields or buttons Enter a name for the object. Enter a description of the object in the Description text box. Enter a keyboard shortcut in the Shortcut text box.

Movie clips Enter a name for the object. Enter a description in the Description text box. Select Make Child Objects Accessible to expose the objects inside the movie clip to screen readers.

Note: If your application can be described in a simple phrase of text that can be easily conveyed by a screen reader, turn off the Make Children Accessible option for your document, and type in a suitable description.

Specifying advanced accessibility options for screen readers

Flash provides several accessibility authoring features that go beyond simply providing names for objects. In addition to providing descriptions for text or text fields, buttons, or movie clips, and keyboard shortcuts for input text fields or buttons, you can also turn off automatic labeling behavior for your document.

You can choose to hide a selected object from screen readers. For example, you should hide objects that are repetitive or do not convey information. You may also decide to hide accessible objects that are contained inside a movie clip or Flash application, and expose only the movie clip or Flash application itself to screen readers.

Turning off automatic labeling for an object and specifying a name

You can specify a name for an individual object if automatic labeling does not provide the correct information.

To turn off an automatic label for an individual object and specify a name for it:

1 On the Stage, select the button or input text field for which you want to control labeling.

2 Do one of the following:

- Select Window > Property inspector if the inspector is not visible. In the Property inspector, click the Accessibility button.
- Select Window > Other Panels > Accessibility.

3 In the Accessibility panel, select Make Object Accessible (the default setting).

4 Enter a name for the object in the Name text box.

The name is read as the label for the button or text field.

5 To turn off accessibility for the automatic label (and hide it from screen readers), select the text object on the Stage.

6 If the text object is static text, convert it to dynamic text: in the Property inspector, select Dynamic Text from the Text type pop-up menu.

7 In the Accessibility panel, deselect Make Object Accessible.

Hiding an object from the screen reader

You can hide an object from the screen reader simply by turning off accessibility for the object. You should only hide objects that are repetitive or convey no content. When an object is hidden, the screen reader ignores the object.

1 On the Stage, select the button or input text field you want to hide from the screen reader.

2 Do one of the following:

- Select Window > Property inspector if the inspector is not visible. In the Property inspector, click the Accessibility button.
- Select Window > Other Panels > Accessibility.

3 In the Accessibility panel, do one of the following:

- If the object is a movie clip, button, text field, or another object, deselect Make Object Accessible.
- If the object is the child of a movie clip, deselect Make Child Objects Accessible.

Creating a keyboard shortcut

You can create a keyboard shortcut for an object, such as a button, so users can quickly navigate to it without listening to the contents of an entire page. For example, you can create a keyboard shortcut so users can quickly navigate to a menu, a toolbar, the next page, or a submit button.

There are two steps to create a keyboard shortcut:

- Code the ActionScript to create a keyboard shortcut for an object. See "`Key class`" in ActionScript Dictionary Help. If you provide a keyboard shortcut for an input text field or button, you must also use the ActionScript Key class to detect the key the user presses during Flash content playback. See "Capturing keypresses" in ActionScript Reference Guide Help.
- Select the object and add the name of the keyboard shortcut to the Accessibility panel so the screen reader can read it.

Keyboard shortcut functionality also depends on the screen reader software used. Be sure to test your Flash content with multiple screen readers. The key combination Control+F, for example, is a reserved keystroke for both the browser and the screen reader. The arrow keys are also reserved by the screen reader. Generally, you can use the keys 0-9 on the keyboard for keyboard shortcuts. However, even those are increasingly used by screen readers, so it is very important to test your keyboard shortcuts. See "Testing accessible content" on page 336.

To indicate the name of a keyboard shortcut for the screen reader:

1 On the Stage, select the button or input text field for which you want to create a keyboard shortcut.

2 Do one of the following:

- Select Window > Property inspector if the inspector is not visible. In the Property inspector, click the Accessibility button.
- Select Window > Other Panels > Accessibility.

3 In the Shortcut field, type the name of the keyboard shortcut, using the following conventions:

- Spell out key names, such as Control or Alt.
- Use capital letters for alphabetic characters.
- Use a plus sign (+) between key names, with no spaces—for example, Control+A.

Warning: No checking is done by Flash to check that the ActionScript to code the keyboard shortcut has been created.

JPMC-00157924

Keyboard shortcut example

For example, if you want to create a keyboard shortcut, Control+7, for a button with the instance name myButton, you would do the following:

1 Select the object on the Stage, display the Accessibility panel, and in the Shortcut field, type **Control+7**.

2 Enter the following code in the Actions panel:

```
function myOnPress() {
  trace( "hello" );
}

function myOnKeyDown() {
  if (Key.isDown(Key.CONTROL) && Key.getCode() == 55) // 55 is key code for
  7
  {
    Selection.setFocus( myButton );
    myButton.onPress();
  }
}

var myListener = new Object();
myListener.onKeyDown = myOnKeyDown;
Key.addListener( myListener );

myButton.onPress = myOnPress;
myButton._accProps.shortcut = "Ctrl+7"
Accessibility.updateProperties();
```

Note: The example assigns the keyboard shortcut Control+7 to a button with an instance name of myButton, and makes information about the shortcut available to screen readers. In this example, when you press Control+7 the `myOnPress` function displays the text "hello" in the Output panel. See `Key.addListener()` in ActionScript Dictionary Help.

Making an entire Flash application accessible

After a Flash document is complete and ready to be published, make the entire Flash application accessible.

To define accessibility for an entire Flash application:

1 When the Flash document is complete and ready to be published or exported, deselect all elements in the document and do one of the following:
   - Select Window > Property inspector if the inspector is not visible. In the Property inspector, click the Accessibility button.
   - Select Window > Other Panels > Accessibility.

2 In the Accessibility panel, select Make Movie Accessible (the default setting) to expose the document to screen readers.

3 Select or deselect the Make Children Accessible option to expose or omit any accessible objects in the document to screen readers.

4 If you selected Make Movie Accessible in step 3, enter information for the document as needed:
   - Enter a name for the document in the Name text box.
   - Enter a description of the document in the Description text box.

5 Select Auto Label (the default setting) to use text objects as automatic labels for accessible buttons or input text fields contained in the document. Deselect this option to turn off automatic labeling and expose text objects to screen readers as text objects.

Using sound with screen readers

Sound is the most important medium for most screen reader users. Consider how any sound in your document will interact with the text spoken aloud by screen readers. It might be difficult for screen reader users to hear what their screen readers are saying if there is a lot of loud sound in your Flash application.

Viewing and creating tab order and reading order

There are two aspects to tab indexing order—the *tab order* in which a user navigates through the web content and the order in which things are read by the screen reader, called the *reading order*.

Flash Player uses a tab index order from left to right and top to bottom. However, if this is not the order you want to use, you can customize both the tab and reading order using the `tabIndex` property in ActionScript (In ActionScript, the `tabIndex` property is synonymous with the reading order).

Tab order You can create a tab order that determines the order in which objects receive input focus when users press the Tab key. You can use ActionScript to do this, or if you have Flash MX 2004 Professional, you can use the Accessibility panel to specify the tab order. Keep in mind that the tab index that you assign in the Accessibility panel does not necessarily control the reading order. See "Creating a tab order index for keyboard navigation in the Accessibility panel (Flash Professional only)" on page 331.

Reading order You can also control the order in which a screen reader reads information about the object (known as the reading order). To create a reading order, you must use ActionScript to assign a tab index to every instance. You must create a tab index for every accessible object, not just the focusable objects. For example, dynamic text must have tab indexes, even though a user cannot tab to dynamic text. If you do not produce a tab index for every accessible object in a given frame, Flash Player ignores all tab indexes for that frame whenever a screen reader is present, and uses the default tab ordering instead. See "Using ActionScript to create a tab order for accessible objects" on page 335.

Creating a tab order index for keyboard navigation in the Accessibility panel (Flash Professional only)

You can create a tab order index in the Accessibility panel for keyboard navigation. You can create a custom tab order for the following objects:

- Dynamic text
- Input text
- Buttons
- Movie clips, including compiled movie clips
- Components
- Screens

Note: You can also use ActionScript to create a keyboard navigation tab order index. See "Using ActionScript to create a tab order for accessible objects" on page 335.

Tab focus occurs in numerical order, starting from the lowest index number. Once tab focus reaches the highest tab index, focus returns to the lowest index number.

When you move user-set tab indexed objects around in your document, or to another document, Flash retains the index attributes. You should then check for and resolve index conflicts, such as two different objects on the Stage which have the same tab index number.

Caution: If two or more objects have the same tab index in any given frame, Flash follows the order in which the objects were placed on the Stage. It is therefore recommended that you resolve all tab index conflicts to be sure the desired tab order index is achieved.

To create a tab order index:

1 Select the object in which to assign a tab order and do one of the following:
   - Select Window > Property inspector if the inspector is not visible. In the Property inspector, click the Accessibility button.
   - Select Window > Other Panels > Accessibility.

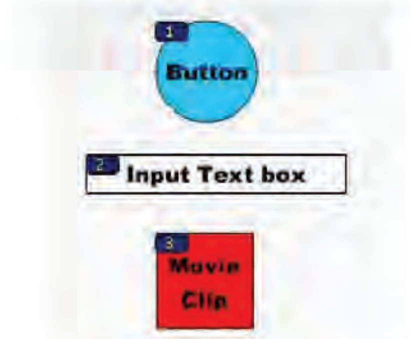
2 If you're providing an index for the selected object only, in the Tab Index text box, enter a positive integer (up to 65535) that reflects the order in which the selected object should receive focus.

   Note: For information about creating a tab order using ActionScript, see "Using ActionScript to create a tab order for accessible objects" on page 335. Tab indexes created in ActionScript do not appear on Stage when the Show Tab Order option is enabled.

To view a tab order:

- Select View > Show Tab Order.

  Tab index numbers for individual objects appear in the upper left corner of the object.



Note: Tab order created with ActionScript code, rather than the Accessibility panel, does not appear when you enable the Show Tab Order option.

About animation and accessibility for the visually impaired

In some situations, you may want to change the property of an accessible object during the course of movie playback. For example, you may want to indicate changes that take place on a keyframe in an animation.

To update properties for an accessible object:

1. Display the frame in which you want to change the properties.
2. Do one of the following:
   - Select Window > Property inspector if the inspector is not visible. In the Property inspector, click the Accessibility button.
   - Select Window > Other Panels > Accessibility.
3. In the Accessibility panel, change the properties for the object as needed.

Alternatively, you can use ActionScript to update accessibility properties. See "Creating accessibility with ActionScript" on page 334.

Different screen readers treat new objects on frames differently. Some screen readers may read only the new object. Some screen readers may re-read the entire document.

Try to avoid animating the text, buttons, and input text fields in your document. If you keep these kinds of objects stable, you reduce the chance of causing a screen reader to emit extra "chatter" that may annoy users. Also, it's best to avoid making your Flash content loop.

If you're using a feature like Text Break Apart to animate text, Flash Player can't determine the actual text content of that text. Other examples of information-carrying graphics include icons and gestural animations. You can remedy problems like this by providing names or descriptions for certain accessible objects within your document, or for the entire Flash application. See "Making an entire Flash application accessible" on page 330. You can also add supplementary text into your document, or shift important content from graphics to text.

Using accessible components

To accelerate building accessible applications, Macromedia has built a core set of UI components. These components automate many of the most common accessibility practices related to labeling, keyboard access, and testing and help ensure a consistent user experience across rich applications. Flash comes with the following set of accessible components:

- SimpleButton
- CheckBox
- RadioButton
- Label
- TextInput
- TextArea
- ComboBox
- ListBox
- Window
- Alert
- DataGrid

Accessible Flash components have special requirements in order to work with screen readers. The components must contain ActionScript that defines their accessible behavior. For information on which accessible components work with screen readers, see the Macromedia Flash Accessibility web page at www.macromedia.com/software/Flash/productinfo/accessibility/.

For general information about components, see Using Components Help.

For each accessible component, you enable the accessible portion of the component with the `enableAccessibility()` command. This command includes the accessibility object with the component as the document is compiled. Because there is no simple way to remove an object after it has been added to the component, these options are disabled by default. It is therefore important that you enable accessibility for each component. This step needs to be done only once for each component; it is not necessary to enable accessibility for each instance of a component for a given document. See Button component,CheckBox component,ComboBox component, Label component, List component, Radio Button component, and Window component in Using Components Help.

Creating accessibility with ActionScript

In addition to the accessibility features included in the Flash user interface, you can create accessible documents with ActionScript. For accessibility properties that apply to the entire document, you can create or modify a global variable called `_accProps`. See `_accProps` in ActionScript Dictionary Help.

For properties that apply to a specific object, you can use the syntax `instancename._accProps`. The value of `_accProps` is an object that can include any of the following properties:

Property	Type	Equivalent selection in the Accessibility panel	Applies to
`.silent`	Boolean	Make Movie Accessible/Make Object Accessible (inverse logic)	Entire documents Buttons Movie clips Dynamic text Input text
`.forceSimple`	Boolean	Make Child Objects Accessible (inverse logic)	Entire documents Movie clips
`.name`	string	Name	Entire documents Buttons Movie clips Input text
`.description`	string	Description	Entire documents Buttons Movie clips Dynamic text Input text
`.shortcut`	string	Shortcut	Buttons Movie clips Input text

Note: Inverse logic means that a value of `true` in ActionScript corresponds to a checkbox that is not selected in the Accessibility panel, and a value of `false` in ActionScript corresponds to a selected checkbox in the Accessibility panel.

Modifying the `_accProps` variable has no effect by itself. You must also use the `Accessibility.updateProperties` method to inform screen reader users of Flash content changes. Calling the method causes Flash Player to re-examine all accessibility properties, update property descriptions for the screen reader, and, if necessary, send events to the screen reader that indicate changes have occurred.

When updating accessibility properties of multiple objects at once, you only need to include a single call to `Accessiblity.updateProperties` (too frequent updates to the screen reader can cause some screen readers to become too verbose).

See `Accessibility.updateProperties()` in ActionScript Dictionary Help.

Implementing screen reader detection with the Accessibility.isActive() method

To create Flash content that behaves in a specific way if a screen reader is active, you can use the ActionScript method `Accessibility.isActive`, which returns a value of `true` if a screen reader is present, and `false` otherwise. You can then design your Flash content to perform in a way that's compatible with screen reader use, such as by hiding child elements from the screen reader. For detailed information, See `Accessibility.isActive()` in ActionScript Dictionary Help.

For example, you could use the `Accessibility.isActive` method to decide whether to include unsolicited animation or not. Unsolicited animation means animation that happens without the screen reader doing anything. This can be very confusing for screen readers.

The `Accessibility.isActive()` method provides asynchronous communication between the Flash content and Flash Player, which means that a slight real-time delay could occur between the time the method is called and the time in which Flash Player becomes active, returning an incorrect value of `false`. To ensure that the method is called correctly, you can do one of the following:

- Instead of using the `Accessibility.isActive()` method when the Flash content first plays, call the method whenever you need to make a decision about accessibility.
- Introduce a short delay of one or two seconds at the beginning of your document to give the Flash content enough time to contact Flash Player.

  For example, you can attach this method with an `onFocus` event to a button. This generally gives the movie enough time to load and you can safely assume a screen reader user will tab to the first button or object on the Stage.

Using ActionScript to create a tab order for accessible objects

In addition to assigning a tab index to objects with the Accessibility panel (see "Creating a tab order index for keyboard navigation in the Accessibility panel (Flash Professional only)" on page 331), you can create the tab order with ActionScript by assigning the `tabIndex` property to the following objects:

- Dynamic text
- Input text
- Buttons
- Movie clips, including compiled movie clips
- Timeline frames
- Screens

If you create a tab order for a frame and you don't specify a tab order for an accessible object in the frame, Flash Player ignores all of the custom tab order assignments. You should, therefore, provide a complete tab order for all accessible objects. Additionally, all objects assigned to a tab order, except frames, must have an instance name specified in the Instance Name text box of the Property inspector. Even items that are not tab stops, such as text, need to be included in the tab order if they are to be read in that order.

Because static text cannot be assigned an instance name, it cannot be included in the list of the `tabIndex` property values. As a result, a single instance of static text anywhere in the movie causes the reading order to revert to the default.

To specify a tab order, you assign an order number to the `tabIndex` property, as in the following example:

```
_this.myOption1.btn.tabIndex = 1
_this.myOption2.txt.tabIndex = 2
```

See `Button.tabIndex`, `MovieClip.tabIndex`, and `TextField.tabIndex` in ActionScript Dictionary Help.

You can also use `tabChildren` or `tabEnabled` methods to assign custom tab order. See `MovieClip.tabChildren`, `MovieClip.tabEnabled`, and `TextField.tabEnabled` in ActionScript Dictionary Help.

Accessibility for hearing-impaired users

To provide accessibility for hearing-impaired users, you can include captions for audio content that is integral to comprehension of the material presented. A video of a speech, for example, would probably require captions for accessibility, but a quick sound associated with a button probably wouldn't.

There are several ways you can add captions to a Flash document including the following:

- By adding text as captions, taking care to ensure the captions are synchronized on the Timeline with the audio.
- Using Hi-Caption Viewer, a component available from Hi Software that works in conjunction with Hi-Caption SE for use with Flash. The white paper titled *Captioning Multimedia with Hi-Caption SE for Use with Macromedia Flash MX* explains how to use Hi-Caption SE and Flash together to create an a captioned document. The white paper is available on the Macromedia website on the Accessibility White Papers page at www.macromedia.com/macromedia/accessibility/whitepapers/. For more information on Hi-Caption SE, see the link on the Macromedia Accessibility Captioning page at www.macromedia.com/macromedia/accessibility/tools/caption.html.

Testing accessible content

When you test your accessible Flash applications, follow these recommendations:

- If you are designing your document to work with screen readers, download several screen readers and test your application by playing it in a browser with the screen reader enabled. Be sure that the screen reader is not attempting to "talk over" places in your document where you have inserted separate audio. Several screen reader applications provide a demonstration version of the software as a free download and you should try as many as you can to ensure compatibility across screen readers.
- If you are creating interactive content, test it and verify that users can navigate your content effectively using only the keyboard. This can be an especially challenging requirement, because different screen readers work in different ways when processing input from the keyboard—meaning that your Flash content might not receive keystrokes as you intended. Be sure to test all keyboard shortcuts.

CHAPTER 18 Printing from SWF Files

You can add printing functionality to your Macromedia Flash MX 2004 and Macromedia Flash MX Professional 2004 document that lets users print from Flash Player. You can use the ActionScript PrintJob class, or you can use the `print()` or `printAsBitmap()` ActionScript funtions. Users can also access the Flash Player context menu and select the Print command there.

Additionally, users can print from a browser, rather than from Flash Player, by selecting a command such as File > Print from the browser window. However, printing from Flash Player directly, rather than from a browser window Print menu, offers several advantages, including the following:

- Users can print all frames or certain frames that you've labeled as printable from Flash Player. Additionally, you can set the print area of a frame.
- You can specify that content print as vector graphics (to take advantage of higher resolution) or as bitmaps (to preserve transparency and color effects).
- The ActionScript PrintJob object improves upon the `print()` and `printAsBitmap()` functions by adding the ability to print dynamically rendered pages as a single print job. The `PrintJob` object also provides the user's printer settings, which can be used to format reports specifically for the user. See "Using the ActionScript PrintJob class" on page 338.
- Flash Player versions earlier than 4.0.25 (Windows) or 4.0.20 (Macintosh) do not support direct printing of frames. Flash Player 7 and later supports the PrintJob class.

Controlling printing

To control what users can print, keep the following in mind as you set up documents and movie clips for printing:

- Adjust the page layout in any frames that you'll designate as printable to match the desired printed output. Using Flash Player, you can print all shapes, symbols, bitmaps, text blocks, and text fields. Levels in a SWF file are not composited on print output.
- The Flash Player printer driver uses the HTML settings for dimension, scale, and alignment in the Publish Settings dialog box. Use these settings to control the print layout.
- The selected frames print as they appear in the movie clip symbol. You can let users print a movie clip that is not visible in a browser by setting the movie clip's `_visible` property to `false` using the Actions panel. Changing the property of a movie clip with the `setProperty` action, tweening, or any transformation tool does not affect how a movie clip prints.
- For a movie clip to be printable, it must be on the Stage or workspace and it must be given an instance name.
- All elements must be fully loaded to print. You can use the movie clip `_framesloaded` property to check whether the printable content is loaded. For more information, see `MovieClip._framesloaded` in ActionScript Dictionary Help.

Supported printers

With Flash Player, you can print to both PostScript and non-PostScript printers. For a list of supported Flash Player printing platforms, see the "Macromedia Flash Player Web Printing FAQ" on the Macromedia website (www.macromedia.com/software/flash/open/webprinting/faq.html).

Using the ActionScript PrintJob class

The ActionScript PrintJob class, in addition to offering improvements to print functionality available with the `print()` method, allows you to also render dynamic content at runtime, prompt users with a single print dialog box, and print an unscaled document with proportions that map to the proportions of the content. This capability is especially useful for rendering and printing external dynamic content, such as database content and dynamic text.

Additionally, with properties populated by the `PrintJob.start()`function, your document can access your user's printer settings, such as page height, width, and orientation, and you can configure your document to dynamically format Flash content that is appropriate for the printer settings.

Building a print job

To build a print job, you use functions that complete the tasks in the order outlined below. The sections that follow the procedure provide explanations of the functions and properties associated with the PrintJob object.

Because you are spooling a print job to the user's operating system between your calls to `PrintJob.start()` and `PrintJob.send()`, and because the PrintJob functions might temporarily affect the Flash Player internal view of onscreen Flash content, you should implement print-specific activities only between your calls to `PrintJob.start()` and `PrintJob.send()`. For example, the Flash content should not interact with the user between `PrintJob.start()` and `PrintJob.send()`. Instead, you should expeditiously complete formatting of your print job, add pages to the print job, and send the print job to the printer.

To build a print job:

1. Create an instance of the print job object: `new PrintJob()`.
2. Start the print job and display the print dialog box for the operating system: `PrintJob.start()`. For more information, see "Starting a print job" on page 340.
3. Add pages to the print job (call once per page to add to the print job): `PrintJob.addPage()`. For more information, see "Adding pages to a print job" on page 340.
4. Send the print job to the printer: `PrintJob.send()`. For more information, see "Sending the print job to the printer" on page 342.
5. Delete the print job: `delete PrintJob`. For more information, see "Deleting the print job" on page 343.

Following is an example of ActionScript that creates a print job for a button:

```
myButton.onRelease = function()
{
  var my_pj = new PrintJob();
  var myResult = my_pj.start();
  if(myResult){
    myResult = my_pj.addPage (0, {xMin : 0, xMax: 400, yMin: 0,
      yMax: 400});
    myResult = my_pj.addPage ("myMovieClip", {xMin : 0, xMax: 400,
      yMin: 400, yMax: 800},{printAsBitmap:true}, 1);
    myResult = my_pj.addPage (1, null,{printAsBitmap:false}, 2);
    myResult = my_pj.addPage (0);

    my_pj.send();
  }
  delete my_pj;
}
```

Only one print job may be running at any given time. A second print job can not be created until one of the following has happened with the previous print job:

- The print job was entirely successful and `PrintJob.send()` method was called.
- The `PrintJob.start()` method returned a value of `false`.
- The `PrintJob.addPage()` method returned a value of `false`.
- The `delete PrintJob` method has been called.

Starting a print job

Calling the `PrintJob.start()` method prompts Flash Player to spool the print job to the user's operating system and also prompts the user's operating system print dialog box to appear.

If, from the print dialog box, the user selects an option to begin printing, the `PrintJob.start()` method returns a value of `true`. (The value is `false` if the user cancels the print job, in which case the script should only call `delete`). If successful, the `PrintJob.start()` method sets values for the `paperHeight`, `paperWidth`, `pageHeight`, `pageWidth`, and `orientation` properties.

Depending on the user's operating system, an additional dialog box might appear until spooling is complete and the function `PrintJob.send` is called: calls to `PrintJob.addPage()` and then `PrintJob.send()` should be made expeditiously. If ten seconds elapse between the `PrintJob.start()` function call and the function call `PrintJob.send()`, which sends the print job to the printer, Flash Player effectively calls `PrintJob.send()`, causing any pages that are added using `PrintJob.addPage()` to be printed and spooling to stop.

When a new print job is constructed, the `PrintJob()` properties are initialized to 0. When `PrintJob.start()` is called, after the user selects the print option in the operating system print dialog box, Flash Player retrieves the print settings from the operating system. The `PrintJob.start()` function populates the following properties:

Property	Type	Unit	Notes
`PrintJob.paperHeight`	number	points	Overall paper height.
`PrintJob.paperWidth`	number	points	Overall paper width
`PrintJob.pageHeight`	number	points	Height of actual printable area on the page; does not include any user-set margins
`PrintJob.pageWidth`	number	points	Width of actual printable area on the page; does not include any user-set margins
`PrintJob.orientation`	string	n/a	Portrait or Landscape orientation

Note: A point is a print unit of measurement that is equal in size to one pixel, a screen unit of measure. For more information about unit equivalencies, see "About scaling" on page 342.

Adding pages to a print job

You add pages to your print job with the `PrintJob.addPage()` method. Although the method can include up to four parameters, the only required parameter is `target/level`. The three optional parameters are `printArea`, `options`, and `frameNum`.

If you are not using a particular optional parameter but are using other optional parameters, use NULL in place of the excluded optional parameter.

With all four parameters, the function uses the following syntax:

```
MyPrintJob.addPage(target[,printArea:Object, options:Object,
   frameNum:Number]):boolean;
```

If you provide an invalid parameter, the print job uses default parameter values, which are specified in the sections that follow.

Each call to add a new page is unique, which allows you to modify parameters without affecting previously set parameters. For example, you can specify that one page print as a bitmap image, and another page print as a vector graphic. You can add as many new pages to your print job as the print job requires. One call to add a page equals one printed page.

Note: Any ActionScript that needs to be called to change a resulting printout must run before the `PrintJob.addPage()` method is called. The ActionScript can, however, run before or after a new `PrintJob()`. If a frame has a call to the `PrintJob.addPage()` method, the call itself does not guarantee that the ActionScript script on that frame will run when that frame is printed.

Specifying a target

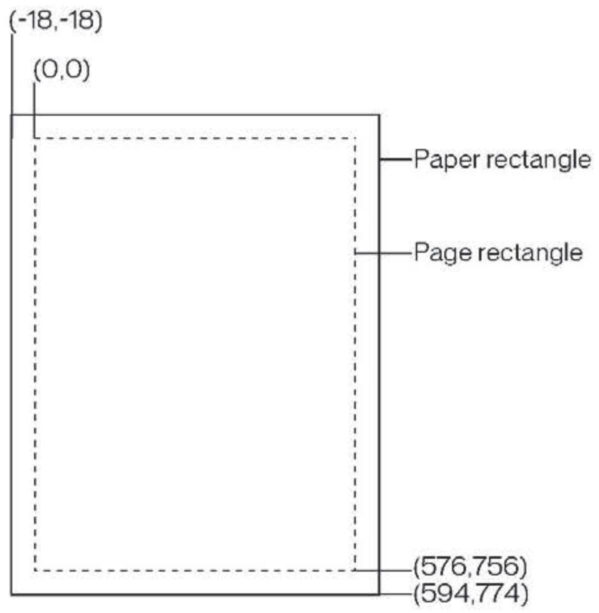
The `target` parameter can be either a number that represents a level (such as 0 for the _root document), or a string that represents the instance name of a movie clip (`"myMovieClip"`).

Specifying a print area

The `printArea` optional parameter includes the following values:

```
{xMin:Number, xMax:Number, yMin:Number, yMax:Number}
```

The `xMin`, `xMax`, `yMin`, and `yMax` values represent screen pixels relative to the target level or movie clip registration point. The print area orientation is from the upper left corner of the printable area on the page. If the print area is larger than the printable area on the page, then the print data that exceeds the right and bottom edge of the page is clipped.



If you don't specify a print area, or if you specify an invalid print area, the print area defaults to the Stage area of the root document.

About scaling

A print job using the PrintJob class prints Flash content, by default, without scaling it. For example, an object that is 144 pixels wide on screen will print as 144 points, or 2 inches wide (One point equals one pixel. In the authoring tool, 72 pixels equals one inch; on paper, 72 points equals one inch.)

To understand how Flash screen content maps to the printed page, it helps to understand screen and print units of measure. Pixels are a screen measurement and points are a print measurement. Both pixels and points equal 1/72 of an inch. A *twip* is 1/20 of a point and pixel.

The following list further illustrates the relationship between units of measure.

- 1 pixel = 20 twips
- 1 point = 20 twips
- 72 pixels = 1 inch
- 72 points = 1 inch
- 567 twips = 1 cm
- 1440 twips = 1 inch

To scale a movie clip before printing, set its `MovieClip.xscale` and `MovieClip.yscale` properties before calling this method, then set them back to their original values afterward. If you scale a movie clip and also pass a value for the *`printArea`* property, the pixel values passed to *`printArea`* reflect the original size of the movie clip. That is, if you set a movie clip's scale to 50% and specify a print area of 500 x 500 pixels, the content that prints is identical to the content that would print if you didn't scale the movie clip; however, it prints at half the size. For more information, see `PrintJob.addPage()` in ActionScript Dictionary Help.

Specifying printing as a vector image or bitmap graphic

The `options` parameter lets you specify whether to print as a vector graphic or bitmap image. When using this optional parameter, use the following syntax:

```
{printAsBitmap:boolean}
```

The default value is `false`, which represents a request for vector printing. Keep in mind the following suggestions when determining which value to use:

- If the content that you're printing includes a bitmap image, then specify that the print job print as a bitmap to include any transparency and color effects.
- Conversely, if the content does not include bitmap images, then specify that the print job print as vector graphics to take advantage of the higher image quality.

Specifying a frame to print

The `frameNum` parameter lets you specify a frame to print. If you do not specify a frame number parameter, the current frame of the target or level specified as the first parameter when adding a page prints by default.

Sending the print job to the printer

To send the print job to the printer after using the `addPage()` calls, use the `PrintJob.send()` method, which causes Flash Player to stop spooling the print job so that the printer starts printing.

Deleting the print job

After sending the print job to a printer, use the ActionScript function `delete PrintJob` to delete the PrintJob object and free memory. For more information, see `delete` in ActionScript Dictionary Help.

Printing frames independent of the PrintJob class

The PrintJob class, available for Flash Player 7 and later, offers many advantages over the `print()` and `printAsBitmap()` methods for printing. However, to print targeting Flash Player 6 and earlier versions, back to Flash Player 4.0.25 (Windows) and 4.0.20 (Macintosh), you can use `print()` and `printAsBitmap()` functions and frame labels—classic functionality that remains part of the authoring tool and does not use the PrintJob class.

To set up printing from Flash Player independent of the PrintJob class, you can specify frames to print and set their print area.

For more information on use of the PrintJob class, see "Using the ActionScript PrintJob class" on page 338.

Designating printable frames (when not using the PrintJob object)

All frames in the specified Timeline print by default. You may want to limit the number of frames that print—for example, if you have a lengthy animation of dozens of frames. You can designate specific frames in a SWF file as printable in order to print only those frames; unspecified frames won't print.

To specify frames as printable, you label the frames.

To designate printable frames:

1. Open or make active the SWF file that you want to publish.
2. Select the desired frame in the Timeline that you want to make printable, and add a keyframe.
3. In the Property inspector (Window > Properties), enter **#p** in the Label text box to specify the frame as printable.
4. Repeat steps 2 and 3 for each frame you want to designate as printable.

   Note: If you have multiple #p labels in your document, you might receive an Output window message when you test or publish your SWF file that indicates that the document contains duplicate frame labels. You can ignore the message if the duplicate labels are all #p labels.

To control what users can print, keep the following in mind as you set up documents and movie clips for printing:

- Adjust the page layout in any frames that you'll designate as printable to match the desired printed output. Using Flash Player, you can print all shapes, symbols, bitmaps, text blocks, and text fields. Levels in a SWF file are not composited on print output.
- The Flash Player printer driver uses the HTML settings for dimension, scale, and alignment in the Publish Settings dialog box. Use these settings to control the print layout.
- The selected frames print as they appear in the movie clip symbol. You can let users print a movie clip that is not visible in a browser by setting the movie clip's `_visible` property to `false` using the Actions panel. Changing the property of a movie clip with the Set Property action, tweening, or any transformation tool does not affect how a movie clip prints.
- For a movie clip to be printable, it must be on the Stage or workspace and it must be given an instance name.
- All elements must be fully loaded to print. You can use the movie clip `_framesloaded` property to check whether the printable content is loaded. For more information, see `MovieClip._framesloaded` in ActionScript Dictionary Help.

Specifying a print area (when not using the PrintJob object)

By default, when frames are printed, the document file's Stage determines the print area. Any object that extends off the Stage is clipped and does not print. Loaded movies use their own Stage size for the print area, not the main movie's Stage size.

As an alternative to using a document's Stage size, you can set the following print areas:

- For either the Flash Player context menu or the `print()` function, you can designate the SWF content bounding box as the print area for all frames by selecting an object in one frame as the bounding box. This option is useful, for example, if you want to print a full-page data sheet from a web banner.
- With the `print()` function, you can use the composite bounding box of all printable frames in a Timeline as the print area—for example, to print multiple frames that share a registration point. To use the composite bounding box, you use the `bMax` parameter, as in the following example:
  ```
  print ("myMovie", "bmax")
  ```
- With the `print()` function, you can change the print area for each frame, scaling objects to fit the print area—for example, to have objects of different sizes in each frame fill the printed page. To change the bounding box per frame, use the Frame parameter in the Print action parameters, as in the following example:
  ```
  print ("myMovie", "bframe")
  ```
- With the `print()` function, you can designate the bounding box of a specific frame in a document as the print area for all printable frames in the document, as in the following example:
  ```
  print ("myMovie", "bmovie")
  ```

You use the label #b to designate a frame to be used to designate the print area. The label #b must be on the same layer as a frame labeled #p.

For more information about `print()` function parameters, see `print()` in ActionScript Dictionary Help.

To specify a print area when printing frames:

1 Open the Flash document (FLA) containing the frames you will set to print.

2 Select a frame that you have not specified to print with a #p frame label. Select a frame that is on the same layer as one labeled #p.

   To organize your work, you can select the next frame after one labeled #p.

3 Create a shape on the Stage the size of the desired print area.

   You can also select a frame with any object of the appropriate print area size to use that frame's bounding box.

4 Select the frame in the Timeline that contains the shape you'll use for the bounding box.

5 If the Property inspector is not displayed, select Window > Properties.

6 In the Property inspector, enter **#b** for Label to specify the selected shape as the bounding box for the print area.

   You can enter only one #b label per Timeline. This option is the same as selecting the Movie bounding box option with the Print action.

Using the print() function (when not using the PrintJob object)

The basic syntax for the `print()` function, which you can associate with a button or other trigger in your document to activate printing, is as follows:

```
print (target, "Bounding box");
```

The target parameter specifies the location of the frames that print, and the bounding box parameter specifies the print area.

You can add a `print()` function to a button or other element in your document to let users print Flash content. You assign the `print()` function to a button, frame, or movie clip. If you assign a `print()` function to a frame, the action executes when the playhead reaches the designated frame.

The `print()` function lets you print frames in other movie clips in addition to the main Timeline. Each `print()` function sets only one Timeline for printing, but the action lets you specify any number of frames within the Timeline to print. If you attach more than one `print()` function to a single button or frame, the Print dialog box appears for each action executed. For more information about the `print()` function, see `print()` in ActionScript Dictionary Help.

Changing the printed background color

With Flash Player, you can print the background color set in the Document Properties dialog box. You can change the background color for only the frames to be printed by placing a colored object on the lowest layer of the Timeline being printed.

To change the printed background color:

1 Place a filled shape that covers the Stage on the lowest layer of the Timeline that will print.

2 Select the shape and select Modify > Document. Select a color for the printing background.

This changes the entire document's background color, including that of movie clips and loaded movies.

3 Do one of the following:

- To print that color as the document's background, make sure that the frame in which you placed the shape is designated to print. For instructions, see "Specifying a frame to print" on page 342 or "Using the print() function (when not using the PrintJob object)" on page 345.
- To maintain a different background color for nonprinting frames, repeat steps 2 and 3.Then place the shape on the lowest layer of the Timeline, in all frames that are not designated to print.

Using frame labels to disable printing

If you don't want any of the frames in the main Timeline to be printable, you can label a frame as !#p to make the entire SWF file nonprintable. Labeling a frame as !#p dims the Print command in the Flash Player context menu. You can also remove the Flash Player context menu.

If you disable printing from Flash Player, the user can still print frames using the browser Print command. Because this command is a browser feature, you cannot control or disable it using Flash.

To disable printing in the Flash Player context menu by dimming the Print command:

1 Open or make active the Flash document (FLA file) that you want to publish.

2 Select the first keyframe in the main Timeline.

3 Select Window > Properties to view the Property inspector.

4 In the Property inspector, for Label enter **!#p** to specify the frame as nonprinting.

You need to specify only one !#p label to dim the Print command in the context menu.

Note: Alternatively, you can select a blank frame (rather than a keyframe) and label it **#p**.

To disable printing by removing the Flash Player context menu:

1 Open or make active the Flash document (FLA file) that you want to publish.

2 Select File > Publish Settings.

3 Select the HTML tab and deselect Display Menu.

4 Click OK.

For more information on publishing options, see "Publishing Flash documents" on page 281.

Printing from the Flash Player context menu

You can use the Print command in the Flash Player context menu to print frames from any Flash SWF file.

The context menu's Print command cannot print transparency or color effects and cannot print frames from other movie clips; for more sophisticated printing capabilities, use the PrintJob object or the `print()` function. See "Using the ActionScript PrintJob class" on page 338 and "Using the print() function (when not using the PrintJob object)" on page 345.

To print frames using the Flash Player context menu Print command:

1 Open the document with frames you will print.

The command prints the frames labeled #p using the Stage for the print area or the specified bounding box.

If you haven't designated specific frames to print, all frames in the document main Timeline print.

2 Select File > Publish Preview > Default or press F12 to view your Flash content in a browser.

3 Right-click (Windows) or Control-click (Macintosh) in the Flash content in the browser window to display the Flash Player context menu.

4 Select Print from the Flash Player context menu to display the Print dialog box.

5 In Windows, select the print range to select which frames to print:

- Select All to print all frames if no frames are labeled.
- Select Pages and enter a range to print the labeled frames in that range.
- Select Selection to print the current frame.

6 On the Macintosh, in the Print dialog box, select the pages to print:

- Select All to print the current frame if no frames are labeled or to print all labeled frames.
- Select From and enter a range to print the labeled frames in that range.

7 Select other print options, according to your printer's properties.

8 Click OK (Windows) or Print (Macintosh).

Note: Printing from the context menu does not interact with calls to the `PrintJob` object.

Publishing a document with printable frames

You can publish a Flash document with printable frames to the web using the Publish command to generate the necessary Flash HTML templates. For more information, see "Publishing Flash documents" on page 281.

Users must have Flash Player 4.0.25 (Windows) or 4.0.20 (Macintosh) or later to take advantage of any print functionality you have added and to be able to print the designated frames in Flash. You can set up a detection scheme to check for the proper Flash Player version.

Note: When you use the PrintJob class, users must have Flash Player 7 or later.

CHAPTER 19
Creating E-learning Content

Macromedia Flash MX 2004 and Macromedia Flash MX Professional 2004 learning interactions help you create interactive online instructional (e-learning) courses that run in Flash. Using the Flash learning interactions has many benefits:

- Anyone with a Flash-enabled web browser can use the instructional content you create.
- You can customize the interface to meet your needs. Because you are using Flash, you can create high-quality interfaces that load quickly and look the same on different platforms.
- You can easily add interactions to your online course with the Flash Learning Interaction components, which provide a simple interface for entering data without writing code.
- Each individual Flash learning interaction can send tracking information to a server-side learning management system (LMS) that complies with the Aviation Industry CBT Committee (AICC) protocol or Shareable Content Object Reference Model (SCORM) standards.
- Additionally, the quiz templates track cumulative results from a sequence of interactions and can pass them along to the LMS using an enhanced data tracking functionality that conforms to either AICC or SCORM standards.

For a hands-on introduction to using quiz templates and interactions, you can take the tutorials in the eLearning folder which resides in the Flash MX 2004 application folder on your computer.

Getting started with Flash learning interactions

Your e-learning courseware runs on any computer with Flash Player 6 or later and a Flash-enabled web browser.

To track user data from the Flash learning interactions, you must have a web server-side LMS, such as an AICC- or SCORM-compatible system. In addition, users must have Internet Explorer 4.0 or Netscape Navigator 4.0 or later (Windows), or Netscape 4.5 or later (Macintosh). Tracking to an LMS with learning interactions does not work with Internet Explorer on the Macintosh.

About Flash learning interactions

An interaction is a part of a Flash application in which the user interacts with the application to provide a response. A typical response might be answering a question, selecting from the answers True or False, or clicking an area of the screen. You can use the six learning interactions included with Flash to build interactive courseware:

True or False In this type of interaction, the user responds to a question with the answers True or False.

Multiple Choice The user responds to a multiple-choice question.

Fill in the Blank The user types a response that is checked against matching phrases.

Drag and Drop The user responds to a question by dragging one or more onscreen objects to a target.

Hot Spot The user responds by clicking a region (or regions) on the screen.

Hot Object The user responds by clicking an object (or objects) on the screen.

Each of the learning interactions has unique parameters that determine how the interaction appears to the user. The interactions are Flash components, which makes them easy to implement and configure in a Flash document. For additional information about Flash components, see "Customizing Components" in Using Components Help.

Including a Flash learning interaction in a document

You can use either quiz templates or stand-alone interactions in your Flash documents.

- The quiz templates are designed for scenarios in which interaction-based quizzes are required or tracking is necessary. The quiz learning interactions are graphically designed to fit into the quiz format. The quiz templates contain a mechanism that counts a cumulative score and starts and stops the necessary tracking in both AICC- and SCORM-compliant APIs. See the following section.
- The stand-alone interactions are designed for scenarios that require a single interaction, or a series of interactions that need to fit into a specific layout within a Flash document. These are available from the common library and are graphically designed for stand-alone use. You can track the results for each individual stand-alone interaction and submit them to an AICC-compatible LMS. See "Adding learning interactions to a quiz template" on page 356.

To initialize SCORM tracking, you must use a quiz template.

Using the quiz templates

Each of the three quiz templates that come with Flash has a different graphical look and feel, but they are otherwise identical. They all contain the following elements:

- A Welcome page
- One of each of the six learning interaction types
- A Results page
- Navigation elements
- ActionScript to gather AICC and SCORM tracking information

The quiz templates provide built-in navigation to move between interactions. They also include ActionScript that can pass tracking information to a web server.

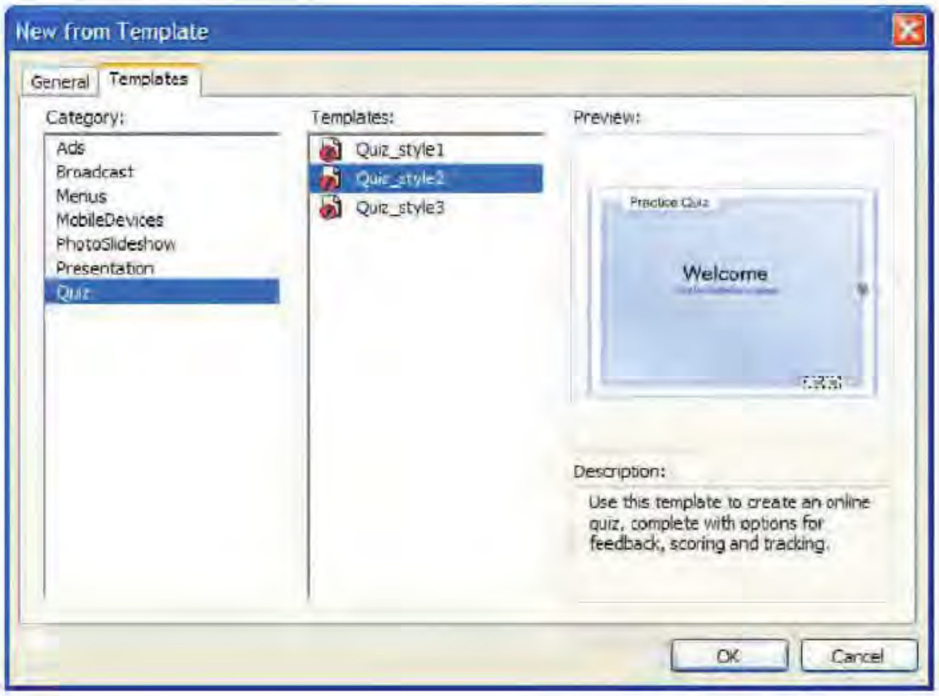
The quiz templates are fully functional. After creating a new document from a quiz template, you can immediately test the document, before modification, to see how the quiz functions. Included with a quiz are each of the six learning interaction types that are stored in movie clips in the library. These movie clips are simply containers for the collection of elements that make up each interaction. You break apart the movie clips to edit the pieces.

To create a quiz:

1 Create a new file by selecting File > New.

2 In the New from Template window, select the Templates tab.

3 In the Category column, select Quiz; then in the Templates column, select one of the quiz styles.



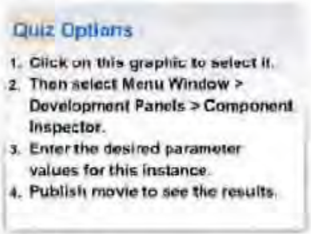
Use this template to create an online quiz, complete with options for feedback, scoring and tracking.

Setting the quiz parameters

After you create a new file and select one of the quiz templates, the next step is to set the quiz parameters. These parameters control how the entire quiz is presented to users—for example, whether the questions are presented in a random or sequential order, the number of question to display, and whether the Results page is displayed.

To set quiz parameters:

1 Select the Quiz Options component with instructions, to the left of the Stage in the quiz template. The component allows you to set the parameters for the quiz.



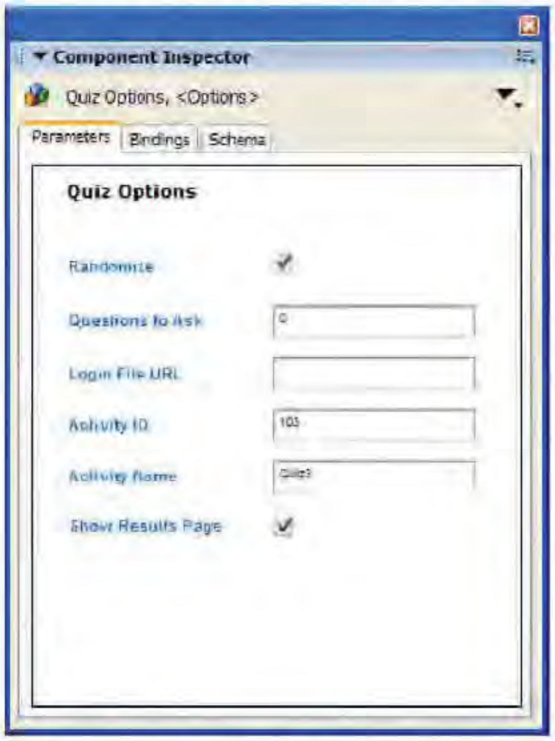
Quiz Options

1. Click on this graphic to select it.
2. Then select Menu Window > Development Panels > Component Inspector.
3. Enter the desired parameter values for this instance.
4. Publish movie to see the results.

Note: These instructions do not appear in SWF file.

2 Do one of the following to open the Component Inspector panel:

- Select Window > Development Panels > Component Inspector.
- In the Property inspector, click Launch Component Inspector.



Quiz Options

Randomize	✓
Questions to Ask	0
Login File URL	
Activity ID	103
Activity Name	Quiz3
Show Results Page	✓

Note: If the text in the Component Inspector panel is too small to be legible, drag a corner of the panel to enlarge it. You may need to undock the panel to enlarge it.

3 Select Randomize if you want the quiz questions to be presented in a random order, not necessarily in the order in which they appear in the Timeline.

4 In the Questions to Ask text box, specify the number of questions to ask for one presentation of the quiz. If you set this number to 0, the quiz uses all the questions you add to the document. If you enter a number larger than the number of questions in the quiz, the quiz displays only the number of questions that are in the quiz and does not duplicate any of them.

For example, if you have 10 interactions in your quiz, you can specify that a lesser number, such as 5 interactions, appear to the user. This feature is especially helpful when used with the Randomize feature to create quizzes with unexpected questions in an unexpected order.

5 Enter the URL to redirect the user.

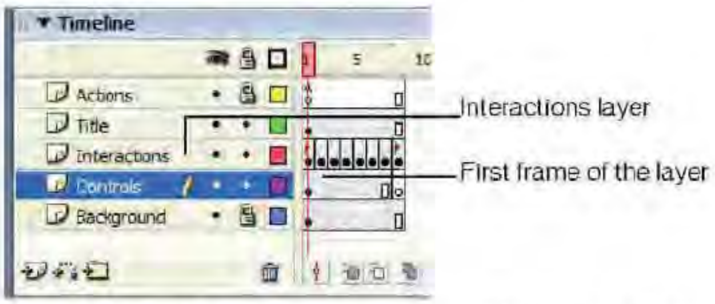
When an AICC-compliant LMS starts a quiz, it includes parameters that the HTML code looks for when it executes the `embed` tag for the Flash application and the course loads properly. If no parameters are specified, the user is redirected to the URL specified in the Login File URL field. If this field is blank or the Flash file was published with the SCORM template, the redirect does not occur.

6 In the Activity ID and Activity Name text boxes, enter the activity ID and activity name of your LMS, if you are using one. If you are not using an LMS, you can either accept or delete the default entries.

7 Select Show Results Page if you want to present quiz results to users after they have completed the quiz.

Modifying learning interactions in a quiz

Each question in the quiz is considered an interaction. When you use a quiz template, you place interactions sequentially between the first and last frame of the Interactions layer on the root Timeline. You may add or remove frames and keyframes as needed, as long as the interactions remain sequential and the first and last frames are reserved for the Welcome page and Results page. The number of frames between the Welcome and Results page keyframes are used to calculate the score.



For example, the following frames would comprise a 10-question quiz:

- Frame 1 = Welcome page keyframe
- Frames 2–11 = interactions keyframes
- Frame 12 = Results page keyframe

These 12 keyframes are on the Interactions layer.

To modify learning interactions in a quiz template:

1 Select the first frame in the Interactions layer and make any modifications you want to the text of the Welcome page. Make sure you include text to indicate that the user must click the Next button to continue. Do not add an interaction to this page.

2 Select each of the learning interactions in the next six frames and do one of the following:

- If you want to use the interaction, follow the instructions in "Configuring a Learning Interaction component" on page 354.
- If you do not want to use the interaction, follow the instructions in "Removing a learning interaction from the Timeline" on page 358.

3 Select the last frame in the Interactions layer and make any modifications you'd like to the text of the Results page. Be careful, however, to leave the supplied dynamic text field names intact, or the results will not appear. Do not delete or place interactions in this frame. If the Results Page quiz parameter is turned off for the quiz, this frame is not called, but it is still reserved.

Configuring a Learning Interaction component

Included with each quiz template is one of each of the six learning interaction types, stored in movie clips in the library. These movie clips are simply containers for the collection of elements that make up each interaction. When you add an interaction (movie clip) to the Stage, you must break it apart to freely edit the individual objects as desired.

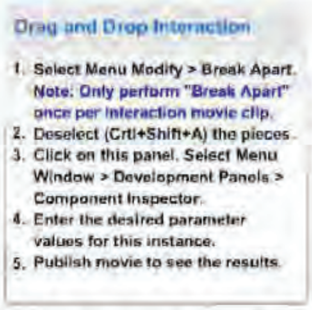
To configure a Learning Interaction component:

1 With the entire learning interaction selected, select Modify > Break Apart. This breaks the interaction into individual objects that can be modified.

Note: Be careful to break apart the interaction only once. See "Testing to see if a movie clip is broken apart" on page 359.

2 Deselect all the items on the Stage (Control+Shift+A).

3 Select the Learning Interaction component.

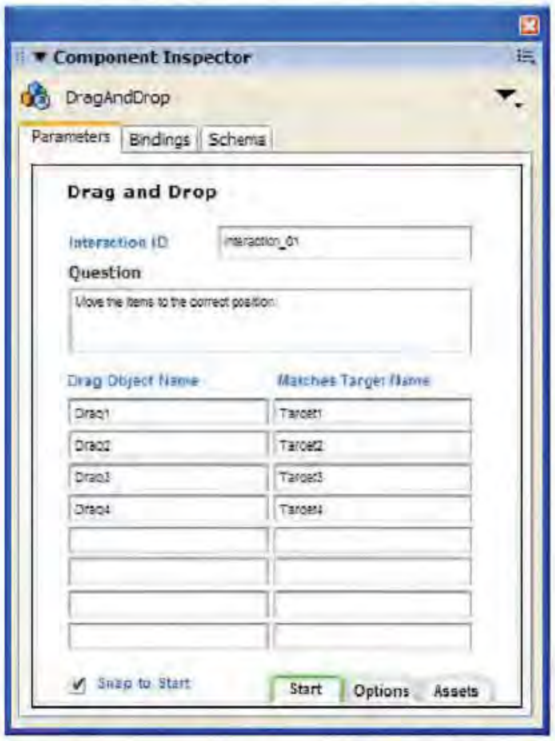


Drag and Drop Interaction

1. Select Menu Modify > Break Apart. Note: Only perform "Break Apart" once per interaction movie clip.
2. Deselect (Crtl+Shift+A) the pieces.
3. Click on this panel. Select Menu Window > Development Panels > Component Inspector.
4. Enter the desired parameter values for this instance.
5. Publish movie to see the results.

Note: Do not delete these instructions from the document; they contain necessary ActionScript code and do not appear in the SWF file.

4 In the Property inspector, click Launch Component Inspector panel.



Drag and Drop

Interaction ID: Interaction_01

Question: Move the items to the correct position.

Drag Object Name	Matches Target Name
Drag1	Target1
Drag2	Target2
Drag3	Target3
Drag4	Target4

✓ Snap to Start

5 If the Flash application will send tracking information to a server-side LMS, specify a name for the interaction in the Interaction ID text box. You should uniquely name each interaction in the quiz as specified by your LMS. Each interaction in the quiz templates is uniquely named. However, if you add interactions from the library or you are not using the quiz template, make sure to uniquely name each interaction in your file.

6 In the Question text box, type the text you want to present to the user. This can be a question and/or instructions for the user.

7 Configure the learning interaction. See the following sections:

- "Configuring a Drag and Drop interaction" on page 361
- "Configuring a Fill in the Blank interaction" on page 363
- "Configuring a Hot Object interaction" on page 364
- "Configuring a Hot Spot interaction" on page 365
- "Configuring a Multiple Choice interaction" on page 367
- "Configuring a True or False interaction" on page 368

8 Click Options, at the bottom of the Component Inspector panel, and enter feedback and Knowledge Track parameters for the learning interaction. See "Adding, naming, and registering assets" on page 368, "Setting Knowledge Track options for a learning interaction" on page 373, and "Setting navigation options for a learning interaction" on page 374.

Note: Documents created using a quiz template have the Knowledge Track option turned on and the Navigation option turned off (the default settings) for each learning interaction, because the quiz template has its own navigation controls.

9 (Optional) Click the Assets button and change the assets for the learning interaction. See "Adding, naming, and registering assets" on page 368.

Adding learning interactions to a quiz template

When you use a quiz template, you add learning interactions to the Interactions layer.

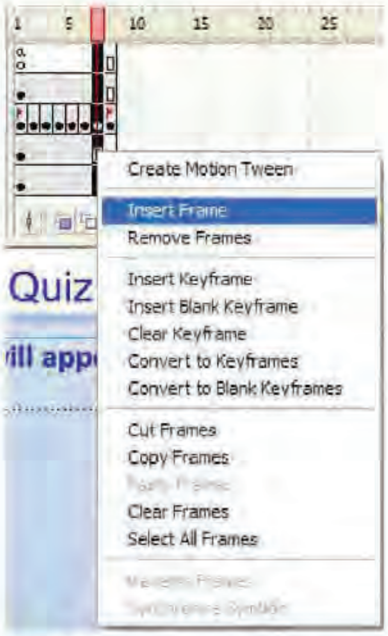
To add an interaction to the Timeline when using a quiz template:

1 In the first layer of the Timeline, select the frame that precedes the frame number in which you want to add the interaction.

For example, if you want to add an interaction to Frame 8, select Frame 7.

2 Shift-click the same frame number on the other layers to select those frames as well.

3 Right-click (Windows) or Control-click (Macintosh) a selected frame and select Insert Frames to extend the Timeline evenly across all layers.



4 On the Interactions layer, select the frame you just added and select Insert > Timeline > Blank Keyframe.

5 To add an interaction, do one of the following:

- To copy and paste an interaction that already exists on the Timeline, right-click (Windows) or Control-click (Macintosh) the keyframe with the interaction and select Copy Frames. Then paste the frame in the blank keyframe that you inserted in step 4. In this copy of the interaction, modify objects on the Stage or the settings in the Component Inspector panel, as desired.
- To use an interaction from the library, drag the desired interaction movie clip type from the Learning Interactions library (Window > Other Panels > Common Libraries > Learning Interactions) to the blank keyframe. Break the interaction apart (select the interaction and select Modify > Break Apart), and edit the assets and parameters.

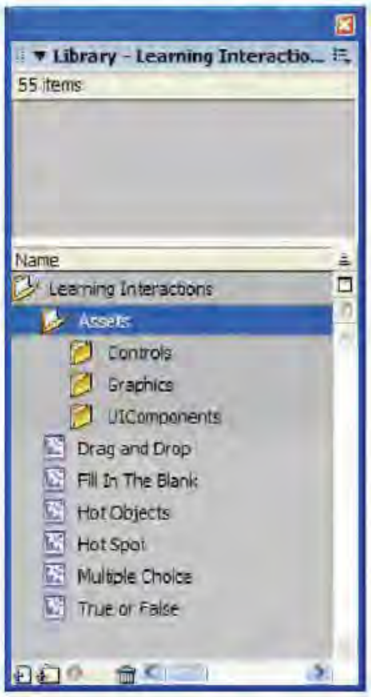
Adding learning interactions to a document that doesn't use a quiz template

If you are adding learning interactions to a Flash document that does not use a quiz template, you can place stand-alone learning interactions on the Timeline in a single frame, sequential frames (for example, 10 questions in 10 sequential frames), or labeled frames.

To add a stand-alone learning interaction to the Timeline when not using a quiz template:

1 If you are adding interactions to a document that does not use the quiz template, select the appropriate layer, then select Insert > Timeline > Blank Keyframe.

2 Select Window > Other Panels > Common Libraries > Learning Interactions. The Learning Interactions library appears.



The library includes six types of learning interaction movie clips: Drag and Drop, Fill in the Blank, Hot Object, Hot Spot, Multiple Choice, and True or False. In addition, there are folders called Assets, Graphics, and UIComponents. These are used for customizing learning interactions. See "Changing buttons, check boxes, and radio buttons" on page 360.

3 Select the new keyframe you created, then drag one of the Learning Interaction movie clips from the Library panel to the Stage.

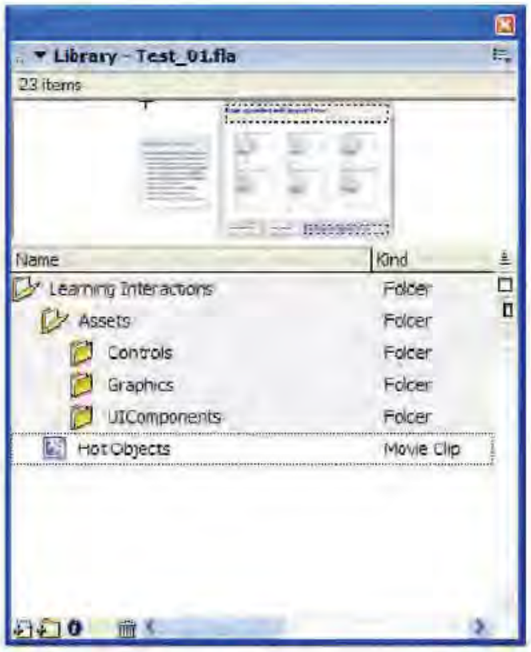
4 Reposition the interaction by dragging it to where you want it to appear on the Stage.

5 Configure the learning interaction. See "Configuring a Learning Interaction component" on page 354.

Note: Be careful to watch the frame count across layers as you are adding and removing keyframes. Make sure that all layers end at the same frame number along the Timeline so that the frame count is the same in all layers.

About managing library assets for learning interactions

When you drag a learning interaction from the Learning Interactions common library to the Stage, the symbols that comprise the learning interaction are copied from the common library to the library of the Flash document you are creating. For example, if you copy a Hot Object learning interaction from the Learning Interactions common library to your document, the symbols in the following illustration become part of the document library.



Name	Kind
Learning Interactions	Folder
Assets	Folder
Controls	Folder
Graphics	Folder
UIComponents	Folder
Hot Objects	Movie Clip

If you're using a quiz template, the learning interaction symbols are already included in your document library.

To manage library assets, it is a good idea to create folders for each graphical interaction and place the folders within the Assets folder. You can then keep the movie clips associated with the interaction within the new folder.

Removing a learning interaction from the Timeline

When you remove a learning interaction from the Timeline, it is important to maintain the sequence of learning interactions. If you remove a frame from the Interactions layer, you need to remove it from all other layers as well.

To remove an interaction from the Timeline:

1 On the Interactions layer, select the keyframe containing the interaction to be deleted. Shift-select the same frame number on other layers if you want to delete those frames as well.

2 To delete frames across all layers, do one of the following:

- Right-click (Windows) or Control-click (Macintosh) the keyframe and select Remove Frames.
- Select Edit > Timeline > Remove Frames.

Note: Be careful to watch the frame count across layers as you are adding and removing keyframes. Make sure that all layers end at the same frame number along the Timeline so that the frame count is the same in all layers.

Testing to see if a movie clip is broken apart

It is a good idea to check whether a learning interaction is broken apart or still grouped within the movie clip container.

To verify whether a learning interaction is broken apart:

- Select a text field or any other single element of the learning interaction on the Stage.

  If a grouped object is selected, the interaction is not broken apart.

  If you can select a single text field or another element, then the interaction has been broken apart and you can proceed with editing.

Changing the appearance of a learning interaction

Once you have added a learning interaction to the Stage and broken it apart, you can place and size most assets just as you would in any other Flash document. For example, you can extend text fields so they can accommodate more lines of text, and you can adjust the font, size, color, and other text properties. Making changes to certain Flash components, such as buttons, check boxes, and radio buttons within learning interactions, however, requires less common processes. See "Changing buttons, check boxes, and radio buttons" on page 360.

Changing the images in a graphical learning interaction

For Drag and Drop, Hot Spot, and Hot Object learning interactions, you change the appearance of the graphic *distractors* (the selectable choices) in the interaction to suit the purposes of your course.

To change the images in a graphical learning interaction:

1. If it has not been broken apart, select the learning interaction movie clip and select Modify > Break Apart.
2. Select the placeholder graphical objects, such as the four placeholder Drag objects and four placeholder Target objects, and delete them.
3. To add your own custom Drag object(s), create or import a graphic and convert it to a movie clip symbol (Modify > Convert to Symbol).
4. Place an instance of the symbol in the desired location on the Stage. In the Property inspector, type the name of the movie clip instance, such as DragA, in the Instance Name text box.
5. In the Component Inspector panel for the interaction, enter the same instance name (such as DragA) of the movie clip in the appropriate Name text box. The Component Inspector panel should include only the unique instance names of the movie clips that you're using for the current interaction.
6. Repeat steps 3–5 for additional graphical objects within the interaction.

Note: The graphics for navigation buttons and for True or False and Multiple Choice interactions are created using Flash UI components. Changing these graphics is recommended only for intermediate and advanced users. For more information, see "Customizing Components" in Using Components Help. You can also resize and slightly modify the appearance of these graphics. See "Changing buttons, check boxes, and radio buttons".

Changing buttons, check boxes, and radio buttons

The learning interactions use the Flash user interface (UI) Button, CheckBox, RadioButton and TextInput components. You must use these UI components within the learning interaction movie clips. The learning interaction scripts use the internal features of the UI components to function properly.

The quiz templates already contain all the necessary UI components for each interaction. To use UI components in Flash MX or later documents, you must publish the SWF file using ActionScript 2.0.

Sizing

The Button components used for the Control button and Reset button can be scaled to fit your needs, as can the CheckBox, RadioButton, and TextInput components.

To set the width and height of the Button, CheckBox, and RadioButton components:

- Select the component and change its settings in the Property inspector.

UI component graphics

There is a defined process for changing the skin of a component. For more information, see "Editing component skins" in Using Components Help.

UI component text

You can use the `GlobalStyleSheet` object to change the text characteristics of a UI component. For detailed information, see "Setting control button labels for a learning interaction" on page 375. See also "Customizing Components" in Using Components Help.

About using components within a learning interaction

To use Flash UI components with a learning interaction, you simply add the UI components to the interaction assets and name their instances. You then need to register the instance names with the component associated with that interaction. Each learning interaction already contains the appropriate UI components as named instances. See "Adding, naming, and registering assets" on page 368.

Complete documentation on the UI components can be found in Flash Help (Help > Using Flash > Using Components).

Note: UI components have a Component Inspector panel associated with them. The learning interaction scripts override the UI Component Inspector panel at runtime. There is no need to fill out individual parameters for each Button, CheckBox, RadioButton or TextInput component.

Testing a quiz

It is important to test a quiz frequently as you add and remove interactions.

To test a quiz:

1 Select Control > Test Movie.

The quiz appears in the Macromedia Flash Player window.

2 Answer the questions as they appear.

3 When you complete the quiz, close it in the Flash Player window to return to the workspace in which you edit the document.

Configuring learning interactions

For each of the six interactions, you must enter specific parameter for the quiz to function properly. A Drag and Drop interaction requires that you specify the Target object and the Drag object. Each Target object and Drag object is referred to as a *distractor*. A distractor is simply one of a series of selectable choices. This term is used for the choices in each of the learning interactions. For example, with a Multiple Choice learning interaction, you enter the multiple-choice distractors.

Configuring a Drag and Drop interaction

You can use up to eight Drag objects and eight Target objects in each Drag and Drop interaction. Each Drag object can snap to any target named in the Drag and Drop component for evaluation. Drag objects can also share targets; for example, both Drag 1 and Drag 2 can match Target 8. You can also specify a target without matching a Drag object to it. This allows you to add incorrect target distractors for evaluation.

Each Target object and Drag object is referred to as a *distractor*. A distractor is simply one of a series of selectable choices. This term is used for the choices in each of the learning interactions.

To configure a Drag and Drop interaction:

1 If you are not using a quiz template, place the learning interaction on the Stage. If you are using a quiz template, select the frame on the Interactions layer that contains the Drag and Drop interaction. (Frame 2, if you have not added or removed keyframes.)

2 Break the movie clip apart (Modify > Break Apart), display the Component Inspector panel, and then type the interaction ID and the question. See "Configuring a Learning Interaction component" on page 354.

3 In the Drag Object Name column, list the instance names for the Drag objects on the Stage.

Each Drag object must have a unique name. If you add a new Drag object on the Stage, be sure to enter its name here.

4 In the Matches Target Name column, list the matching target instance name for that Drag object.

Each target must have a unique name. If you add a new target on the Stage, be sure to enter its name here.

If you enter a Drag instance name in the Drag Object Name column, you need to enter a corresponding Target instance name in the Matches Target Name column. However, you can enter a Target instance name in the Matches Target Name column without a matching Drag instance name. This adds a target that can be snapped to but is not evaluated as a correct match.

5 Select Snap to Start to make the Drag objects snap back to their original position if they do not snap to a registered target.

6 Select each instance of the Drag object or Target object on the Stage. Use the Property inspector to give each instance the same instance name that you specified in the Component Inspector panel.

Adding and removing Drag objects and Target objects

You can change the default number of four objects and four targets by adding additional objects and targets, or by deleting existing objects and targets. You can include from one to eight Drag objects and one to eight Target objects in a Drag and Drop learning interaction.

To add a Drag object or Target object:

1 Create a movie clip symbol containing the graphics for the object. For example, if you have an interaction that has six types of fruit, and you want to add a seventh choice, create a graphic of the seventh fruit and place it in the library.

2 Select the Drag and Drop learning interaction on the Timeline, then drag the symbol from the Library panel to the Stage.

3 In the Property inspector, name the instance. See "Adding, naming, and registering assets" on page 368.

4 Add the instance name to the Component Inspector panel for the Drag and Drop object. See "Naming and registering graphic distractors" on page 370.

The component does the rest of the work for you at runtime.